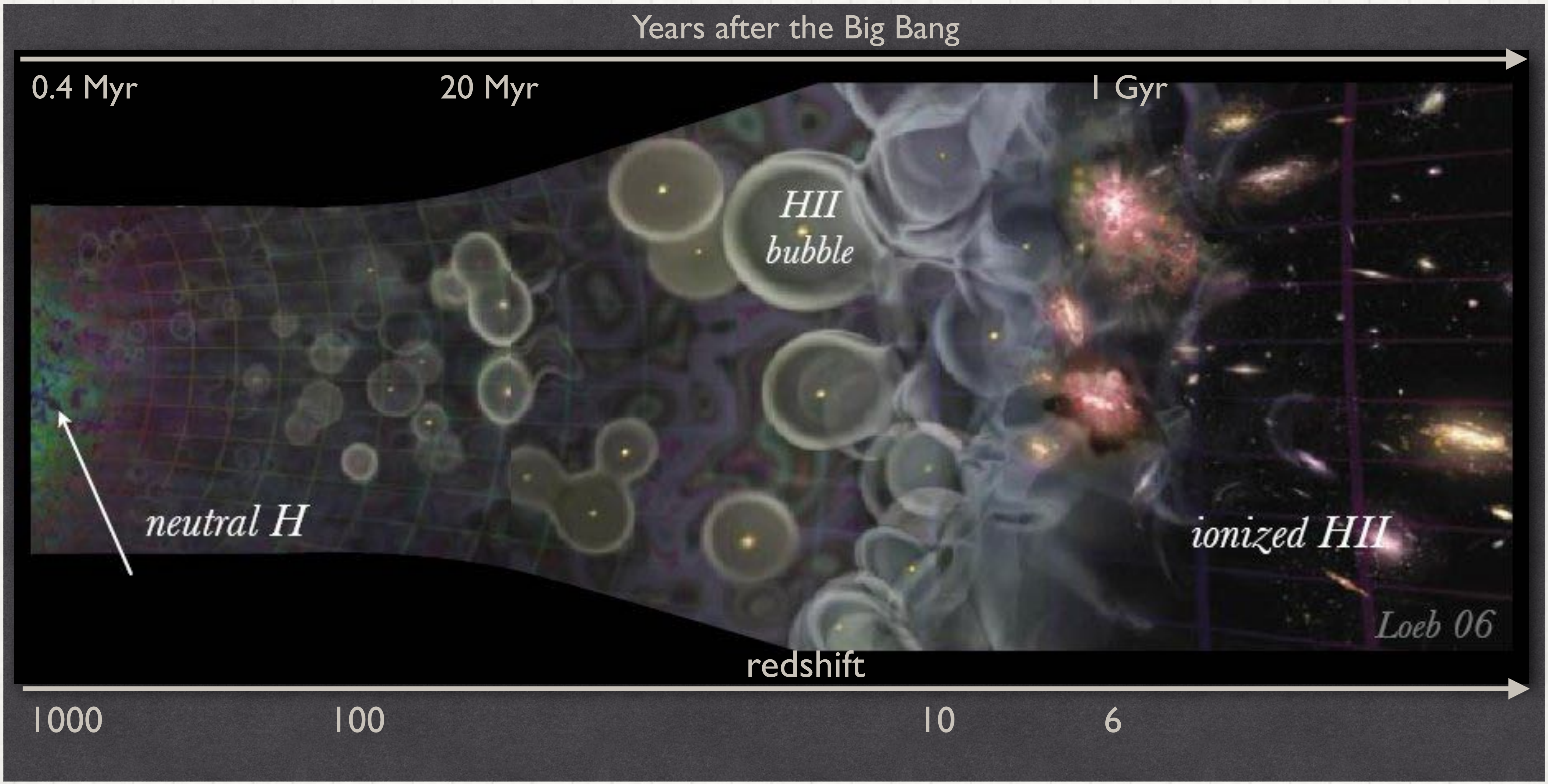
SCIENTIFIC DATA ANALYSIS SCHOOL

EXPLORATION OF 3D SPECTROSCOPY DATA

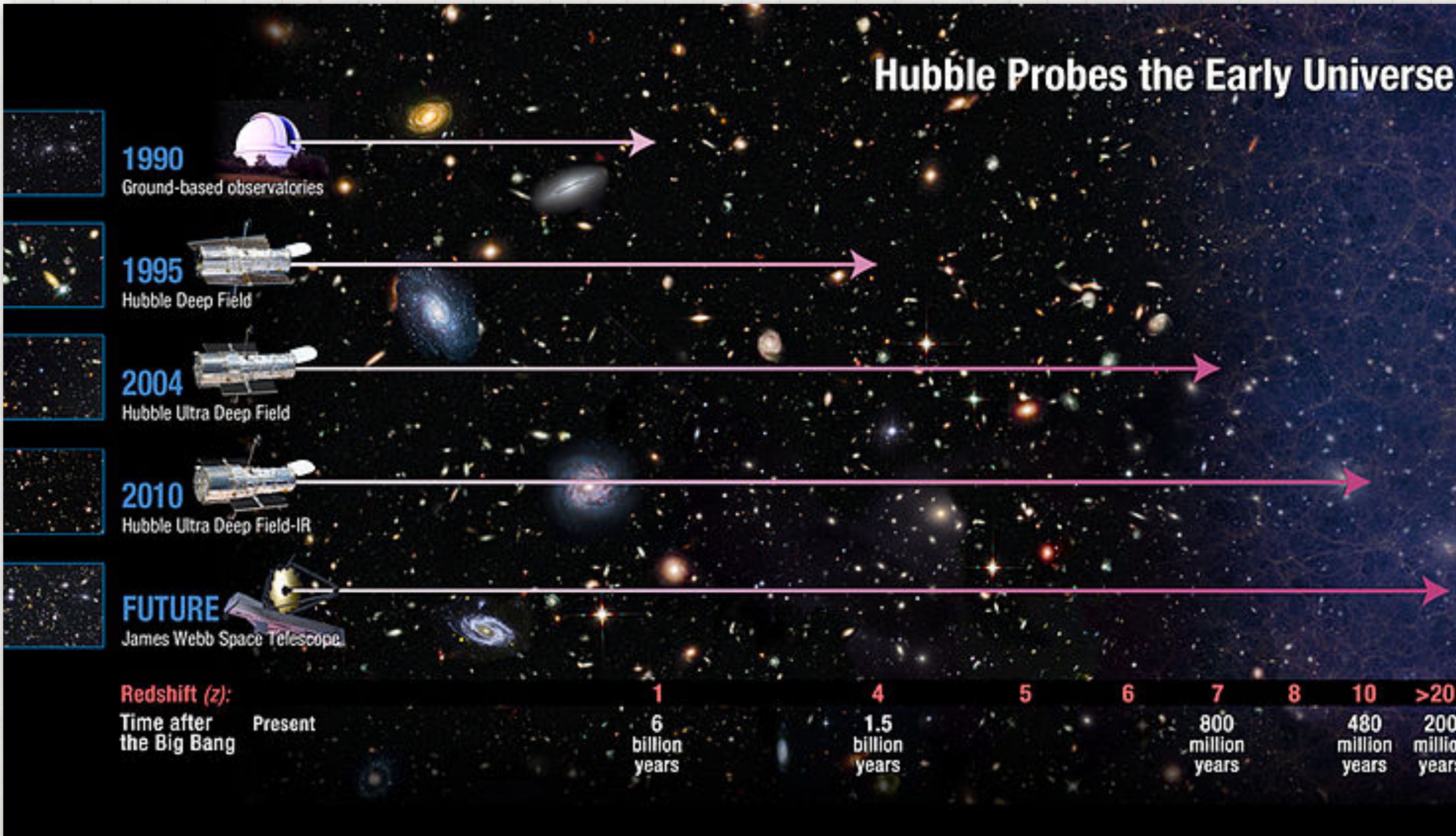
Stefano Carniani
Scuola Normale Superiore

COSMOLOGY

WHEN DID THE FIRST GALAXIES FORM?
HOW DO THEY GROW AND BUILD UP THEIR MASS?



ASTRONOMICAL DATA

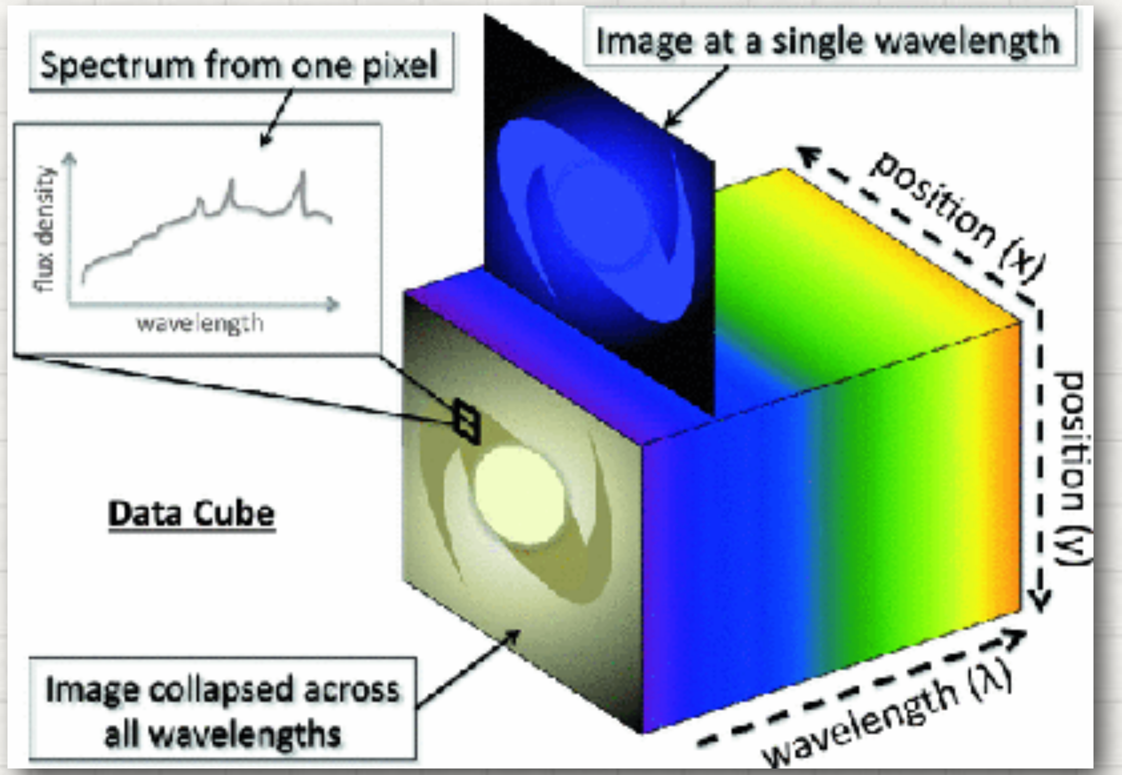


ASTRONOMICAL DATA

IMAGING

LARGE FIELD-OF-VIEW
NO SPECTROSCOPIC INFORMATION

SPECTROSCOPY DATA



SMALL FIELD-OF-VIEW
SPECTROSCOPIC INFORMATION

FITS FILES & DATA VISUALISATION APPLICATION

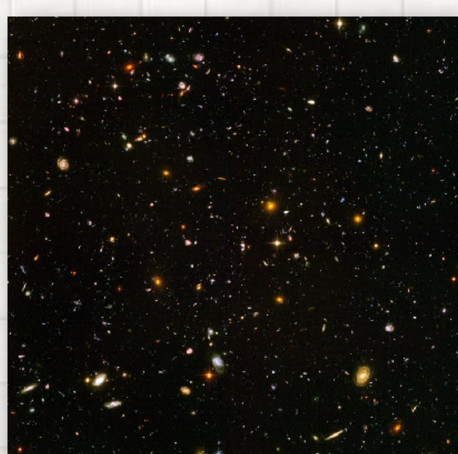
FITS

- "FLEXIBLE IMAGE TRANSPORT SYSTEM" (FITS) IS THE DATA FORMAT MOST WIDELY USED WITHIN ASTRONOMY FOR ARCHIVING AND ANALYSING SCIENTIFIC DATA FILES.

- FITS IS MUCH MORE THAN JUST ANOTHER IMAGE FORMAT (SUCH AS JPG OR GIF) AND IS PRIMARILY DESIGNED TO STORE SCIENTIFIC DATA SETS CONSISTING OF MULTIDIMENSIONAL ARRAYS (IMAGES OR DATA CUBES).

- A FITS FILE IS COMPRISED OF SEGMENTS CALLED `HEADER/DATA UNITS' (HDU), WHERE THE FIRST HDU IS CALLED THE "PRIMARY HDU"

- EVERY HDU CONSISTS OF AN ASCII FORMATTED "HEADER UNIT" FOLLOWED BY AN OPTIONAL "DATA UNIT"

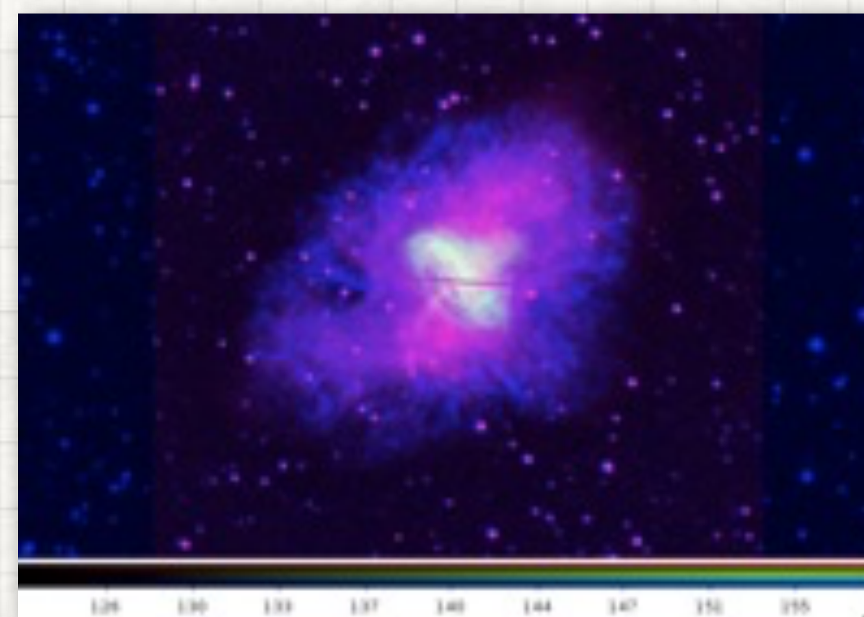


DS9

- DS9 IS AN ASTRONOMICAL IMAGING AND DATA VISUALIZATION APPLICATION.

- DS9 SUPPORTS FITS IMAGESOMPRISED OF SEGMENTS CALLED `HEADER/DATA UNITS' (HDU), WHERE THE FIRST HDU IS CALLED THE "PRIMARY HDU"

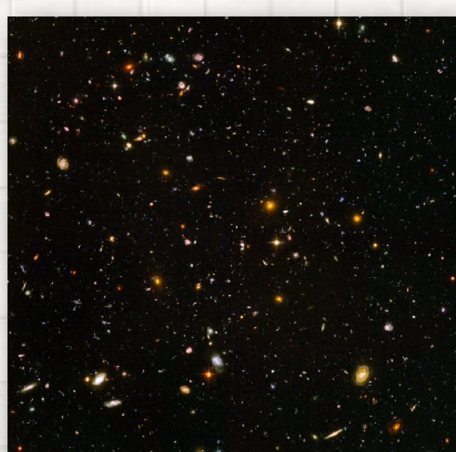
- HTTP://DS9.SI.EDU/SITE/HOME.HTML



FITS FILES & DATA VISUALISATION APPLICATION

FITS

- "FLEXIBLE IMAGE TRANSPORT SYSTEM" (FITS) IS THE DATA FORMAT MOST WIDELY USED WITHIN ASTRONOMY FOR ARCHIVING AND ANALYSING SCIENTIFIC DATA FILES.
- FITS IS MUCH MORE THAN JUST ANOTHER IMAGE FORMAT (SUCH AS JPG OR GIF) AND IS PRIMARILY DESIGNED TO STORE SCIENTIFIC DATA SETS CONSISTING OF MULTIDIMENSIONAL ARRAYS (IMAGES OR DATA CUBES).
- A FITS FILE IS COMPRISED OF SEGMENTS CALLED `HEADER/DATA UNITS' (HDU), WHERE THE FIRST HDU IS CALLED THE "PRIMARY HDU"
- EVERY HDU CONSISTS OF AN ASCII FORMATTED "HEADER UNIT" FOLLOWED BY AN OPTIONAL "DATA UNIT"



DS9

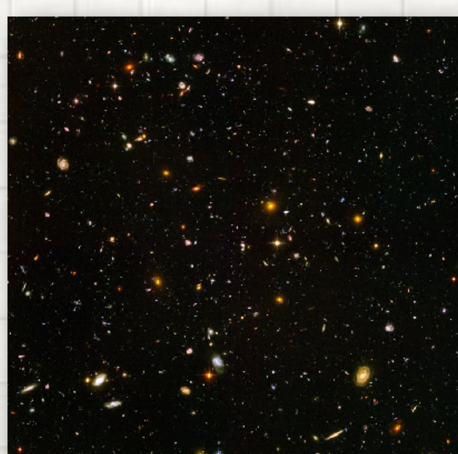
TERMINAL

```
> module load ds9

> ds9 &
```

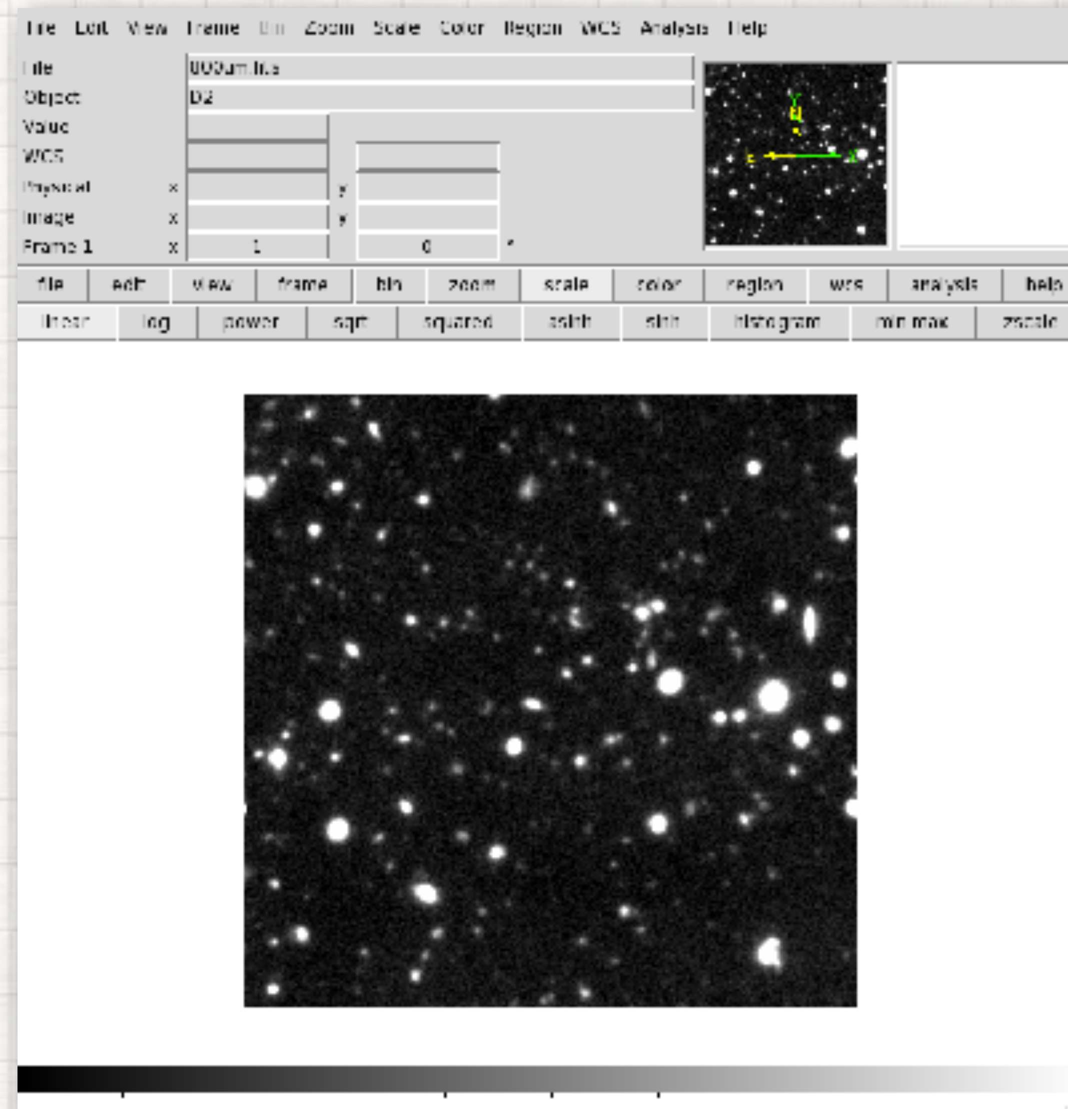
FITS FILES & DATA VISUALISATION APPLICATION

FITS

- “FLEXIBLE IMAGE TRANSPORT SYSTEM” (FITS) IS THE DATA FORMAT MOST WIDELY USED WITHIN ASTRONOMY FOR ARCHIVING AND ANALYSING SCIENTIFIC DATA FILES.
- FITS IS MUCH MORE THAN JUST ANOTHER IMAGE FORMAT (SUCH AS JPG OR GIF) AND IS PRIMARILY DESIGNED TO STORE SCIENTIFIC DATA SETS CONSISTING OF MULTIDIMENSIONAL ARRAYS (IMAGES OR DATA CUBES).
- A FITS FILE IS COMPRISED OF SEGMENTS CALLED `HEADER/DATA UNITS’ (HDU), WHERE THE FIRST HDU IS CALLED THE “PRIMARY HDU”
- EVERY HDU CONSISTS OF AN ASCII FORMATTED “HEADER UNIT” FOLLOWED BY AN OPTIONAL “DATA UNIT”



DS9



FITS FILES & DATA VISUALISATION APPLICATION

FITS

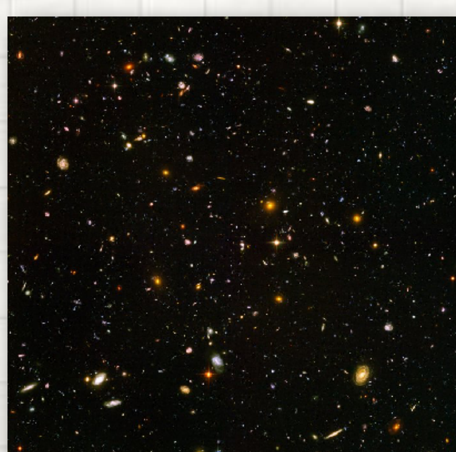
- "FLEXIBLE IMAGE TRANSPORT SYSTEM" (FITS) IS THE DATA FORMAT MOST WIDELY USED WITHIN ASTRONOMY FOR ARCHIVING AND ANALYSING SCIENTIFIC DATA FILES.

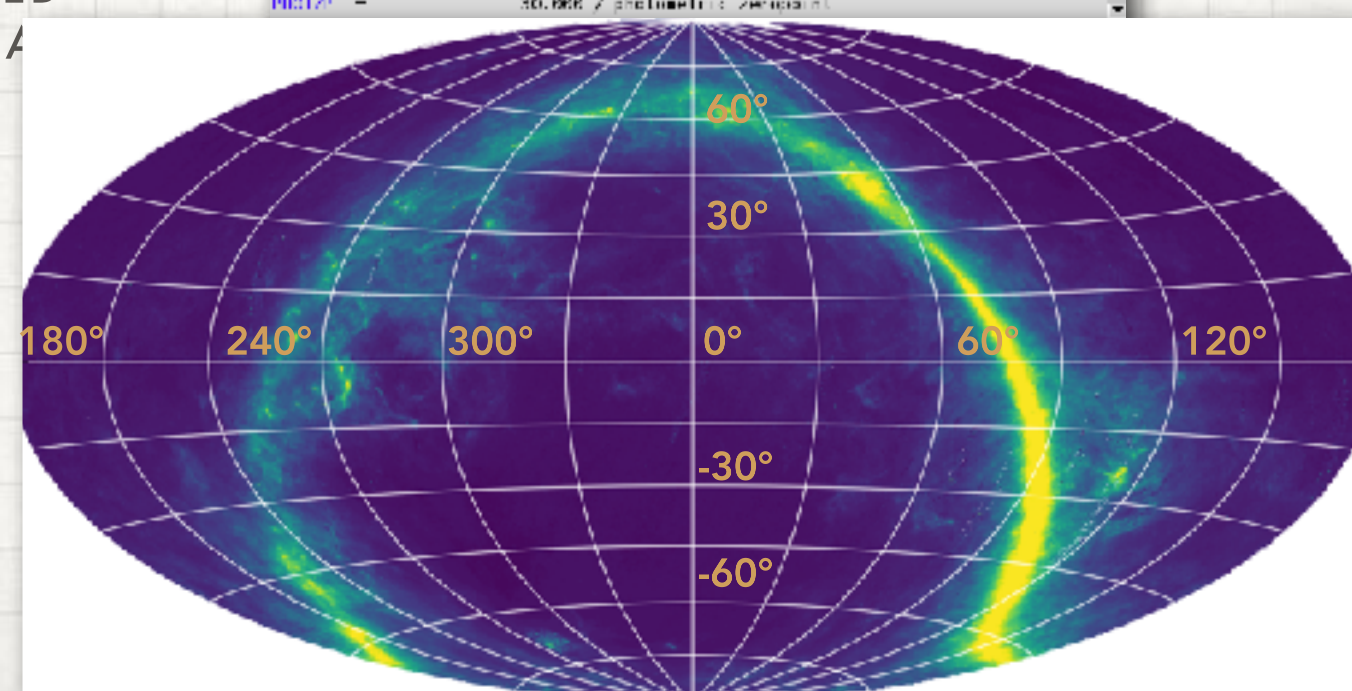
- FITS IS MUCH MORE THAN JUST ANOTHER IMAGE FORMAT (SUCH AS JPG OR GIF) AND IS PRIMARILY DESIGNED TO STORE SCIENTIFIC DATA SETS CONSISTING OF MULTIDIMENSIONAL ARRAYS (IMAGES OR DATA CUBES).

- A FITS FILE IS COMPRISED OF SEGMENTS CALLED `HEADER/DATA UNITS' (HDU), WHERE THE FIRST HDU IS CALLED THE "PRIMARY HDU"

- EVERY HDU CONSISTS OF AN ASCII FORMATTED "HEADER UNIT" FOLLOWED BY AN OPTIONA "DATA UNIT"

DS9





DATA ANALYSIS

WE NEED TO HANDLE WIDE DATA

- There are several public astronomical softwares for data analysis…

- some of current astronomical softwares have been developed for specific scientific goals

- some others astronomical softwares are “black boxes” for users

DATA ANALYSIS

WE NEED TO HANDLE WIDE DATA

- There are several public astronomical softwares for data analysis…

- some of current astronomical softwares have been developed for specific scientific goals

- some others astronomical softwares are “black boxes” for users

a good data analysis requests an appropriate software/code

let’s see how to analyse astronomical data with

astropy
A Community Python Library for Astronomy

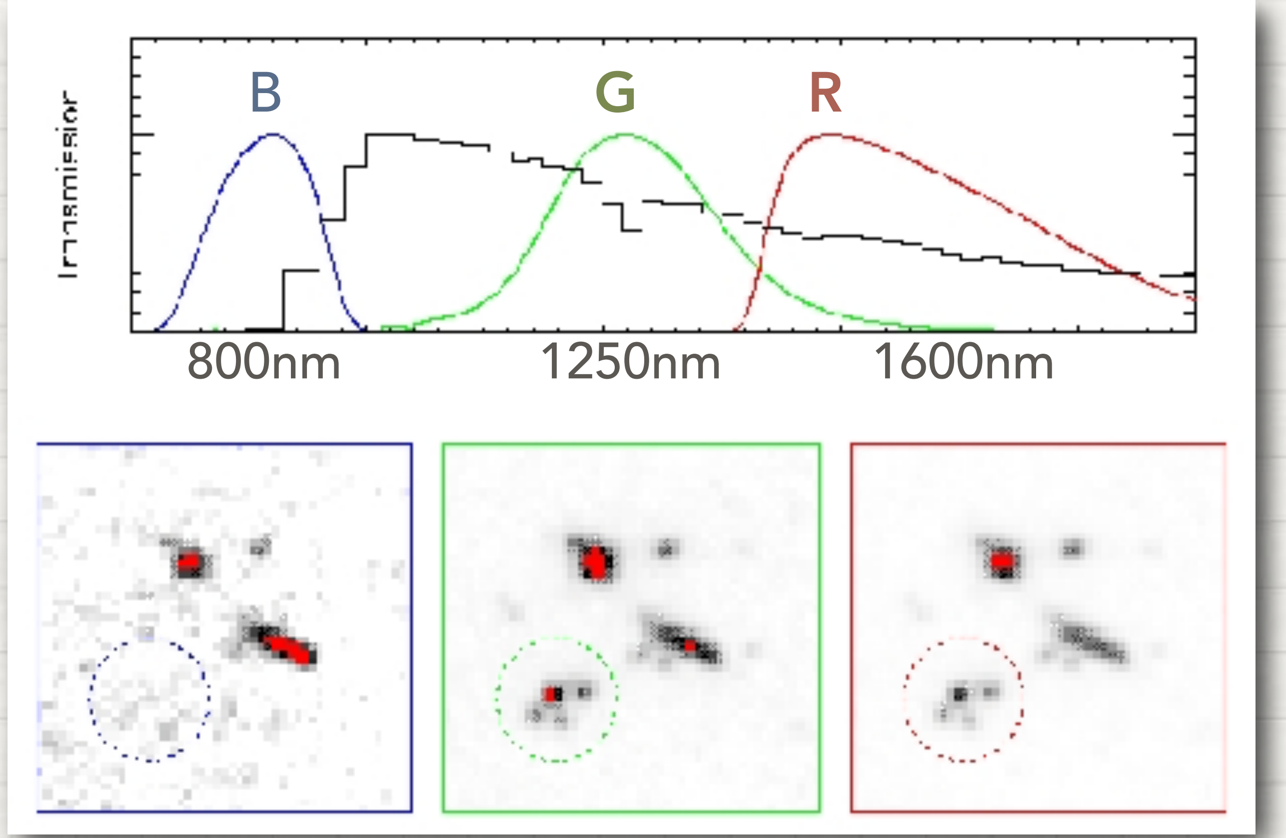
python™

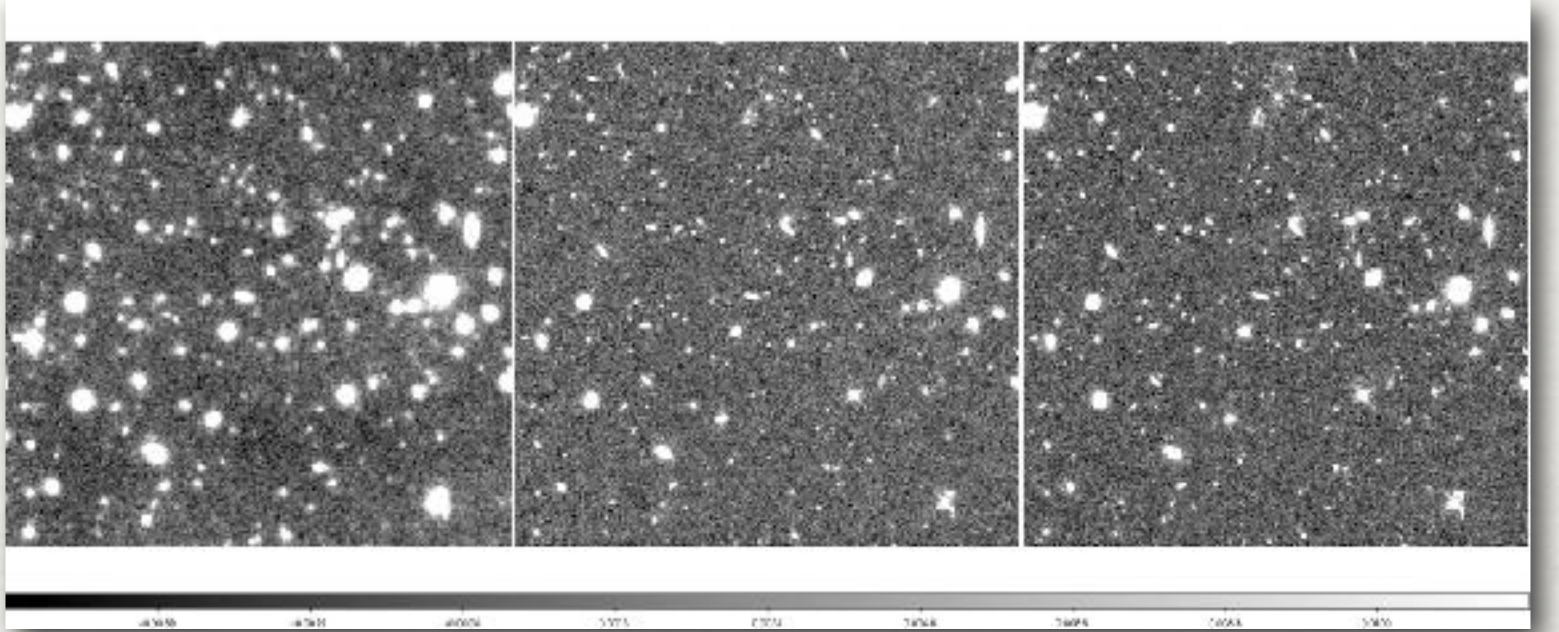
https://www.astropy.org

STEP #1

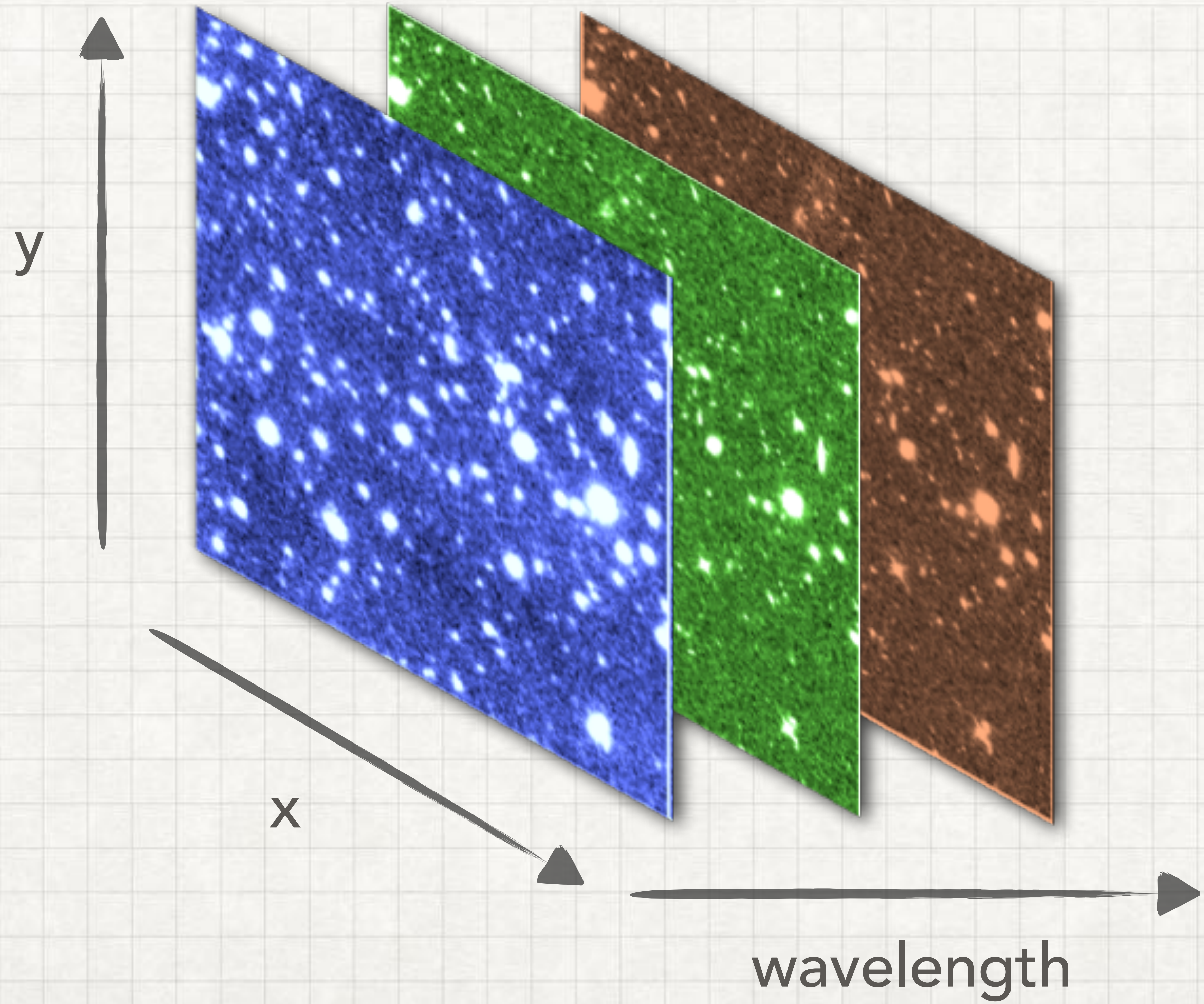
HUNTING GALAXIES IN AN ASTRONOMICAL IMAGE

- The neutral hydrogen clouds in the intergalactic medium along the line of sight to a distant galaxy absorb light in their rest-frame series (i.e. Lyα, Lyβ, … , Lyman limit 912Å)
- we observe a sky field in three different filters
    - blue ~800nm (800um.fits)
    - green ~1250nm (1250um.fits)
    - red ~1600nm (1600um.fits)
- galaxies at z>6 are visible only in the 'green' and 'red' images





EXPLORATION OF 3D DATA



STEP #1

HUNTING GALAXIES IN AN ASTRONOMICAL IMAGE

TERMINAL

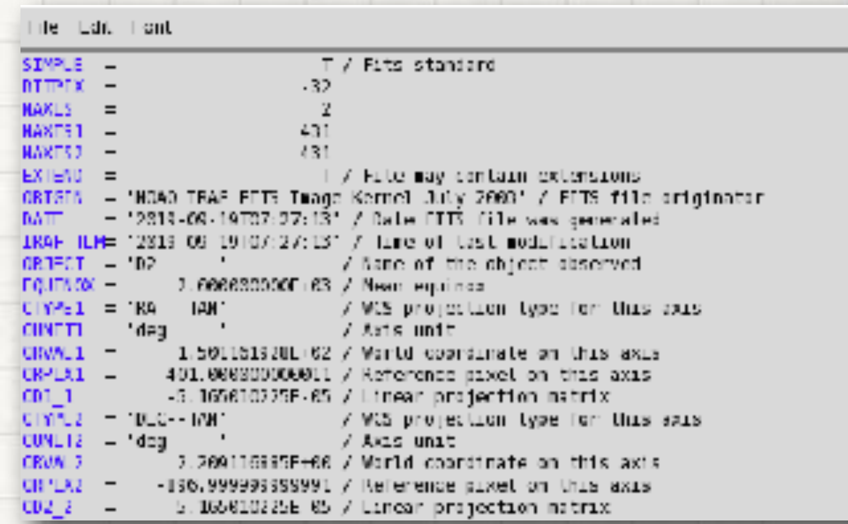
```
> module load python/3.7.2

> df -h

> cd /media/TOSHIBA EXT…

> cd astrophysics

>cd
carniani_exploration_of_3D_spectro
scopy_data

> jupyter-notebook
```

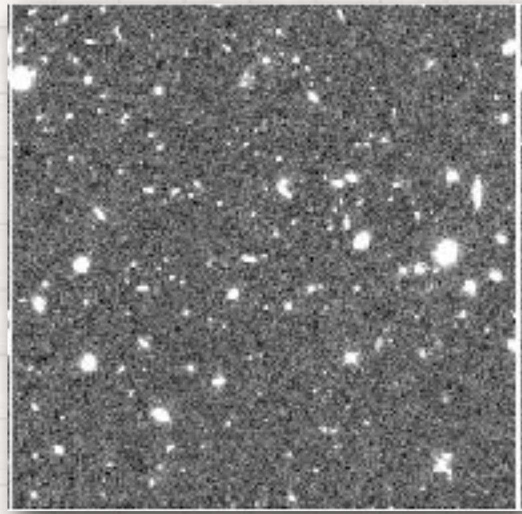
ASTROPY & FITS FILE

PYTHON STRUCTURE

WE SELECT THE FIRST SEGMENT OF FITS FILE

- hst_fits= fits.open(filefits)[0]
- hst_fits.header
- hst_fits.data
- hst_wcs = wcs.WCS(hst_fits.header)





MAPS THE PIXEL LOCATIONS IN A IMAGE TO THE THEIR REAL-WORLD UNITS (I.E. POSITION ON THE SKY SPHERE)

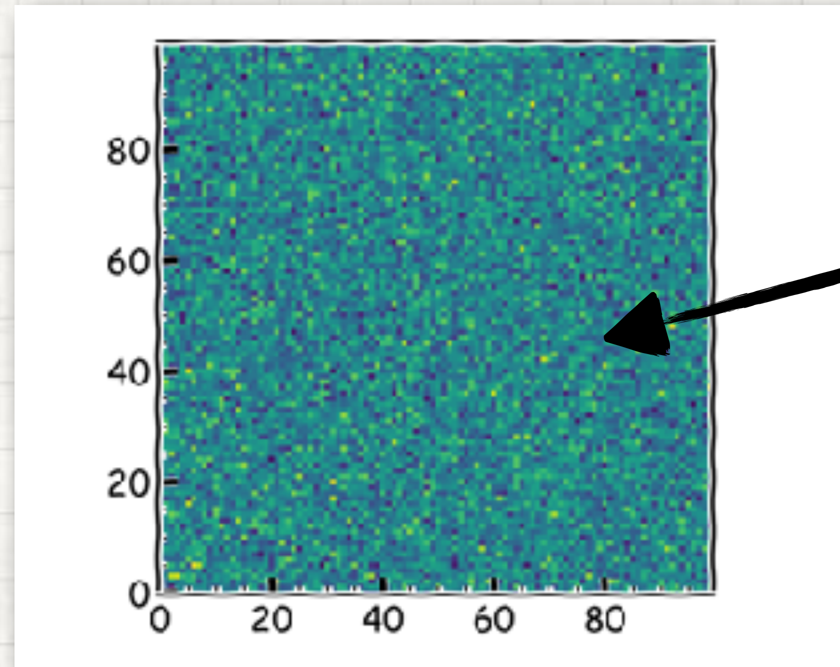
(x,y) -> (RA,Dec)

STEP #1

HUNTING GALAXIES IN AN ASTRONOMICAL IMAGE

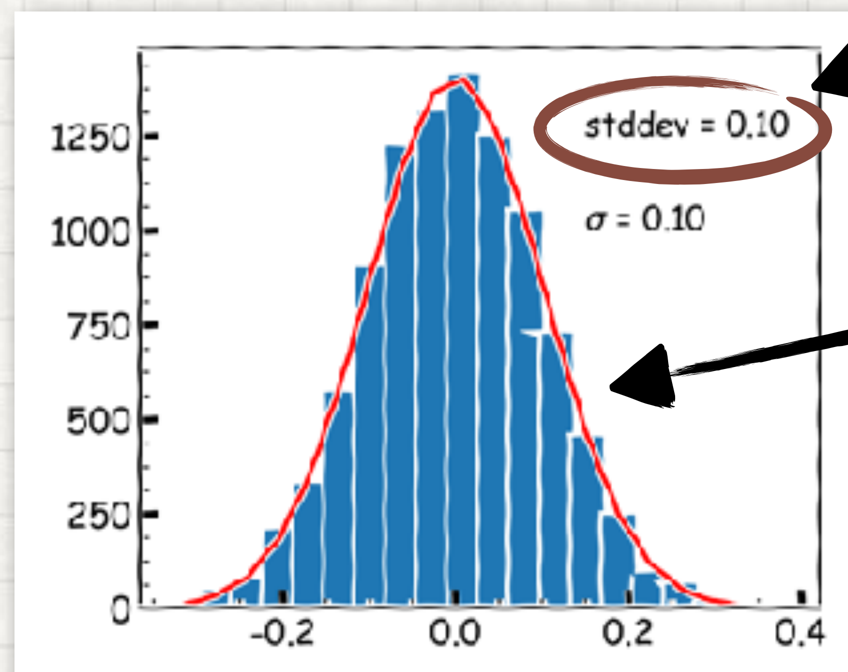
- Determine the noise level in a astronomical image

- Noise usually follows a Gaussian distribution



white noise

$$\sqrt{\frac{\sum (x_i - \bar{x})^2}{N}}$$



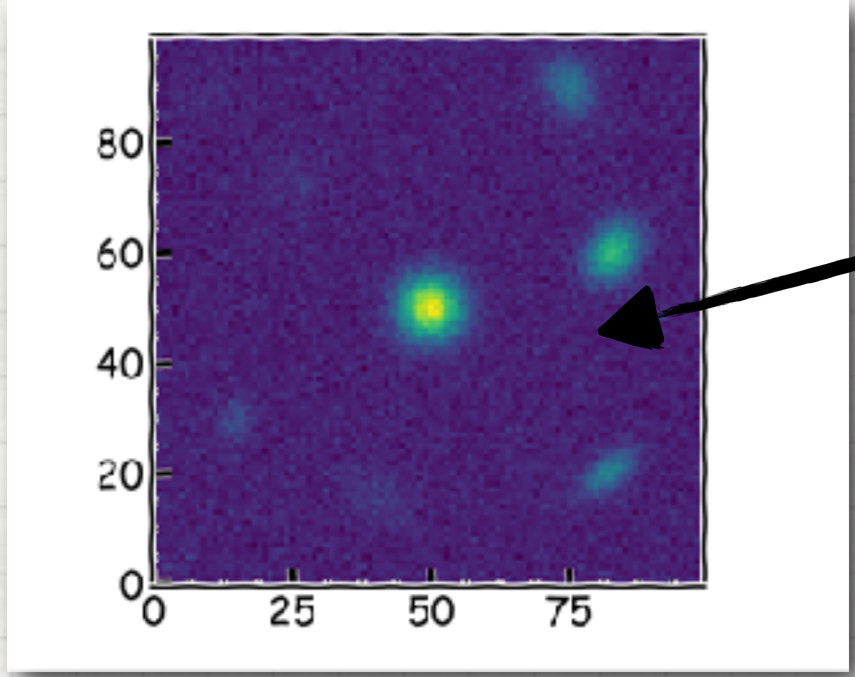
hist. of pixels

```python
image = np.copy(hst_blue.data)
#play with the other data
#image = np.copy(hst_green.data)
#image = np.copy(hst_red.data)


print('min value = {}'.format(np.min(image)))
print('max value = {}'.format(np.max(image)))

time.sleep(0.1)
min_value = float(input('min value of the histogram? '))
max_value = float(input('max value of the histogram? '))
n_bins = int(input("number of bins? "))

bins = np.linspace(min_value, max_value, n_bins)
hist, bins_edge = np.histogram(image,bins = bins, range = [min_value,max_value])
bins_centers = np.array([0.5 * (bins[i] + bins[i+1]) for i in range(len(bins)-1)])

std = np.std(image)
print('standard deviation = {}'.format(std))

g_init = models.Gaussian1D(amplitude=np.max(hist), mean=0, stddev=std/10.,
                           bounds={"stddev": (0.0001, 1)})
#g_init.fixed['mean'] = True
fit_g = fitting.LevMarLSQFitter()
g = fit_g(g_init,bins_centers ,hist)
```

STEP #1

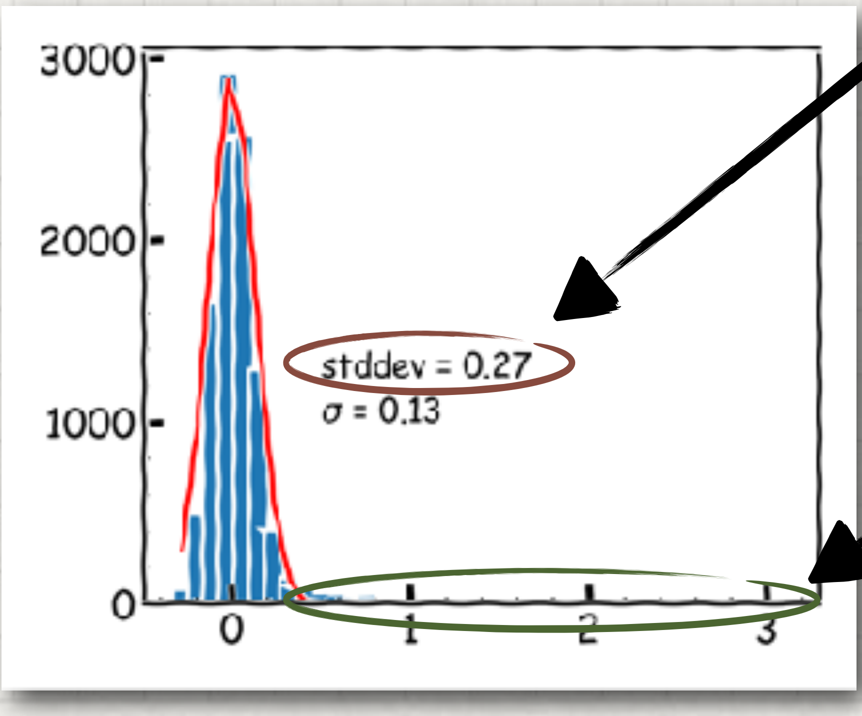
HUNTING GALAXIES IN AN ASTRONOMICAL IMAGE

- Determine the noise level in a astronomical image
- Noise usually follows a Gaussian distribution
- In astronomical observations, we have not any pure noise images



white noise
+
sources

$$\sqrt{\frac{\sum (x_i - \bar{x})^2}{N}}$$



hist. of pixels

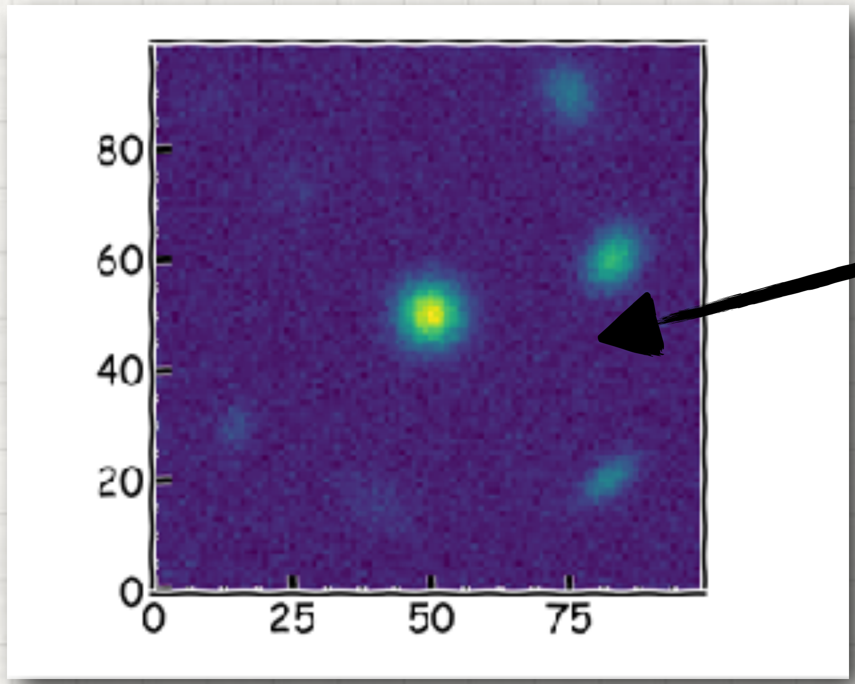
wings due to galaxies

```
image = np.copy(hst_blue.data)
#play with the other data
#image = np.copy(hst_green.data)
#image = np.copy(hst_red.data)


print('min value = {}'.format(np.min(image)))
print('max value = {}'.format(np.max(image)))

time.sleep(0.1)
min_value = float(input('min value of the histogram? '))
max_value = float(input('max value of the histogram? '))
n_bins = int(input("number of bins? "))

bins = np.linspace(min_value, max_value, n_bins)
hist, bins_edge = np.histogram(image,bins = bins, range = [min_value,max_value])
bins_centers = np.array([0.5 * (bins[i] + bins[i+1]) for i in range(len(bins)-1)])

std = np.std(image)
print('standard deviation = {}'.format(std))

g_init = models.Gaussian1D(amplitude=np.max(hist), mean=0, stddev=std/10.,
                           bounds={"stddev": (0.0001, 1)})
#g_init.fixed['mean'] = True
fit_g = fitting.LevMarLSQFitter()
g = fit_g(g_init,bins_centers ,hist)
```

STEP #1

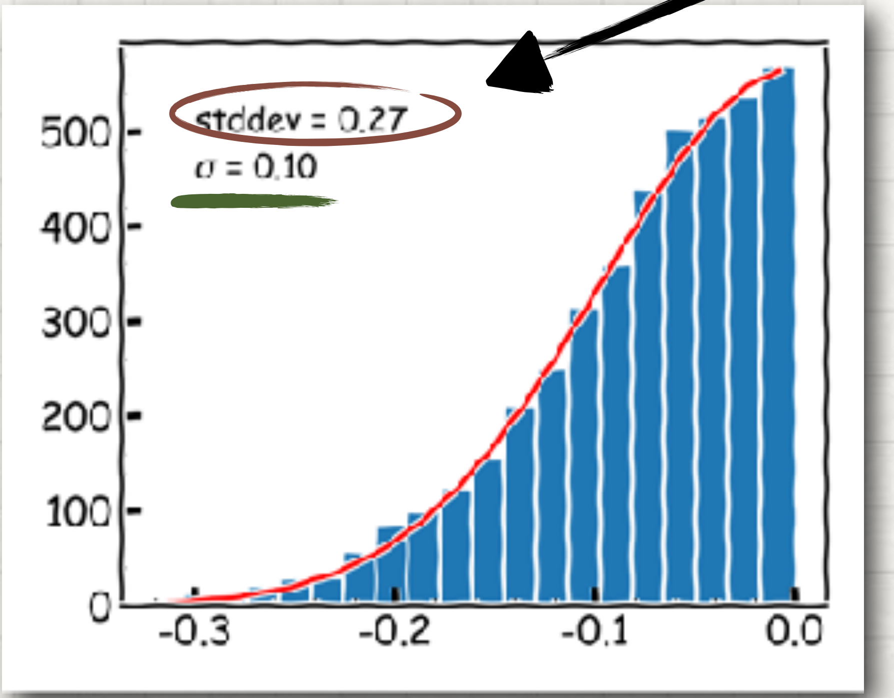
HUNTING GALAXIES IN AN ASTRONOMICAL IMAGE

- Determine the noise level in a astronomical image
- Noise usually follows a Gaussian distribution
- In astronomical observations, we have not any pure noise images
- galaxies have only positive values in astronomical images



white noise
+
sources

$$\sqrt{\frac{\sum (x_i - \overline{x})^2}{N}}$$



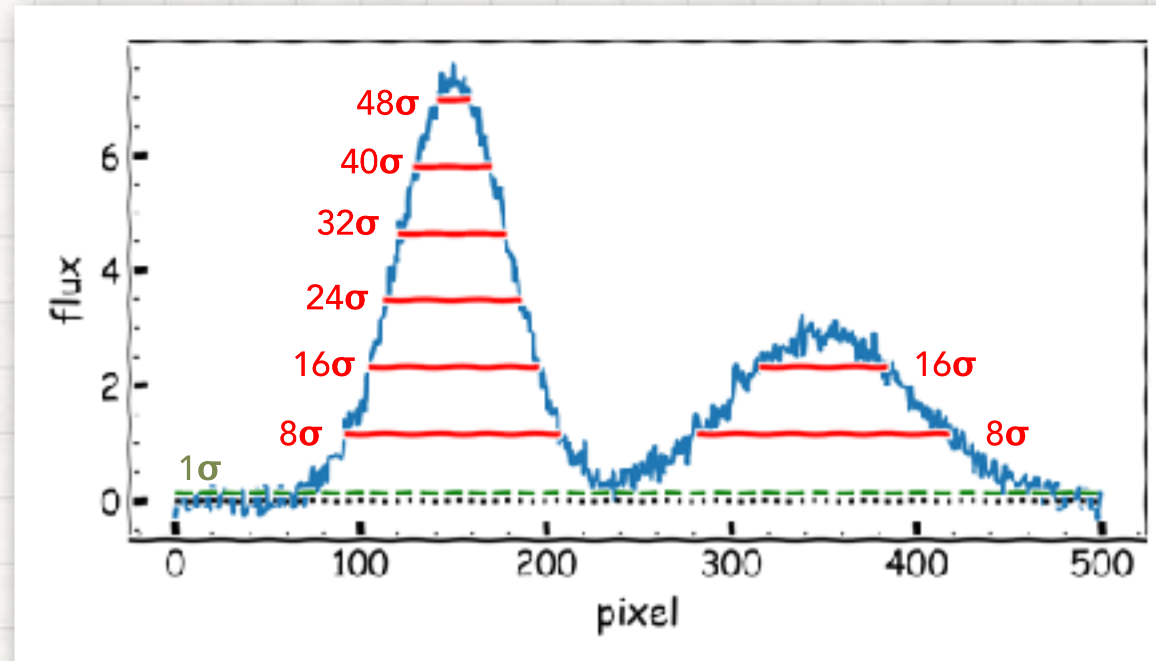
hist. of neg. pixels

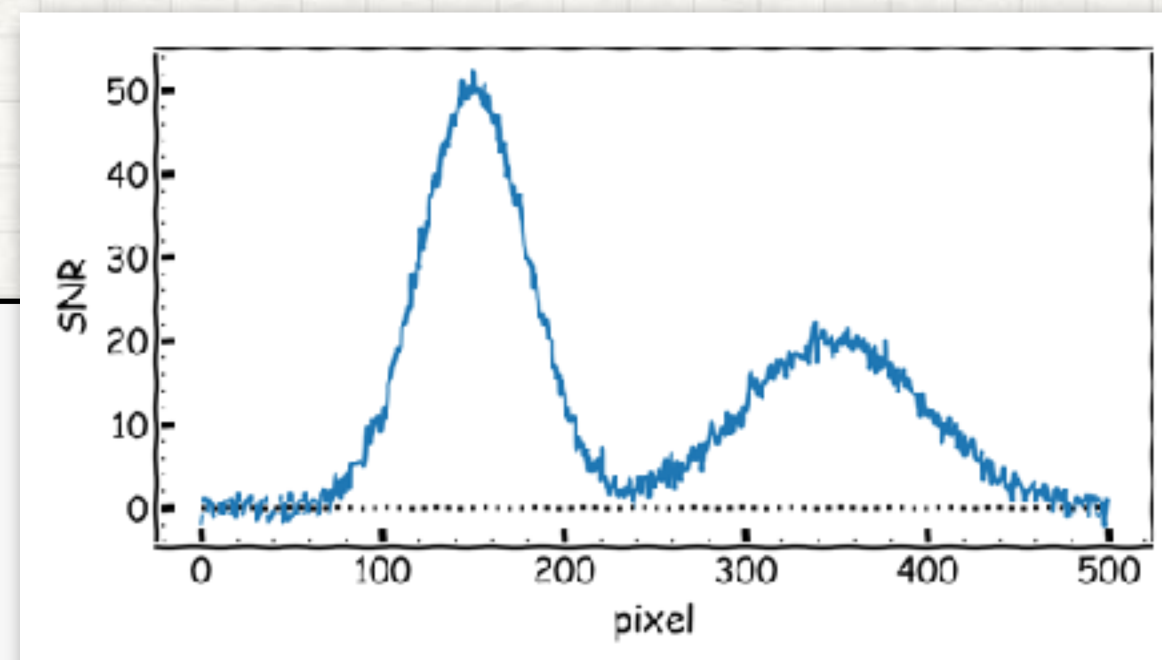
```python
image = np.copy(hst_blue.data)
#play with the other data
#image = np.copy(hst_green.data)
#image = np.copy(hst_red.data)


print('min value = {}'.format(np.min(image)))
print('max value = {}'.format(np.max(image)))

time.sleep(0.1)
min_value = float(input('min value of the histogram? '))
max_value = float(input('max value of the histogram? '))
n_bins = int(input("number of bins? "))

bins = np.linspace(min_value, max_value, n_bins)
hist, bins_edge = np.histogram(image,bins = bins, range = [min_value,max_value])
bins_centers = np.array([0.5 * (bins[i] + bins[i+1]) for i in range(len(bins)-1)])

std = np.std(image)
print('standard deviation = {}'.format(std))

g_init = models.Gaussian1D(amplitude=np.max(hist), mean=0, stddev=std/10.,
                           bounds={"stddev": (0.0001, 1)})
#g_init.fixed['mean'] = True
fit_g = fitting.LevMarLSQFitter()
g = fit_g(g_init,bins_centers ,hist)
```

STEP #1

HUNTING GALAXIES IN AN ASTRONOMICAL IMAGE

- generate signal-to-noise ratio (SNR) map



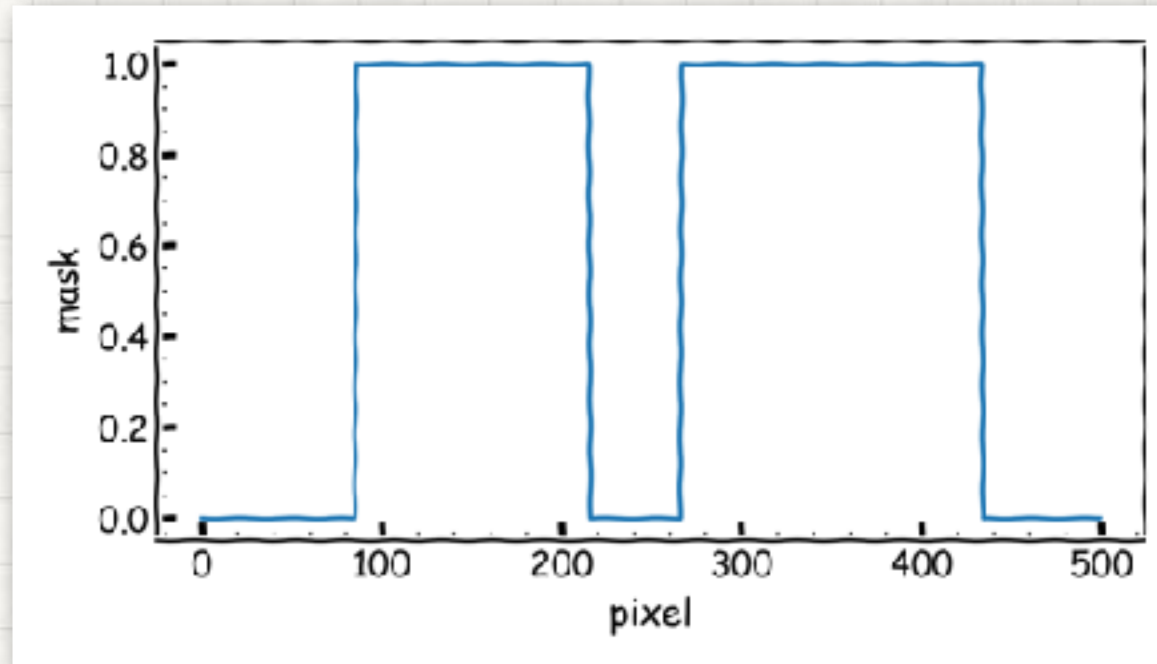


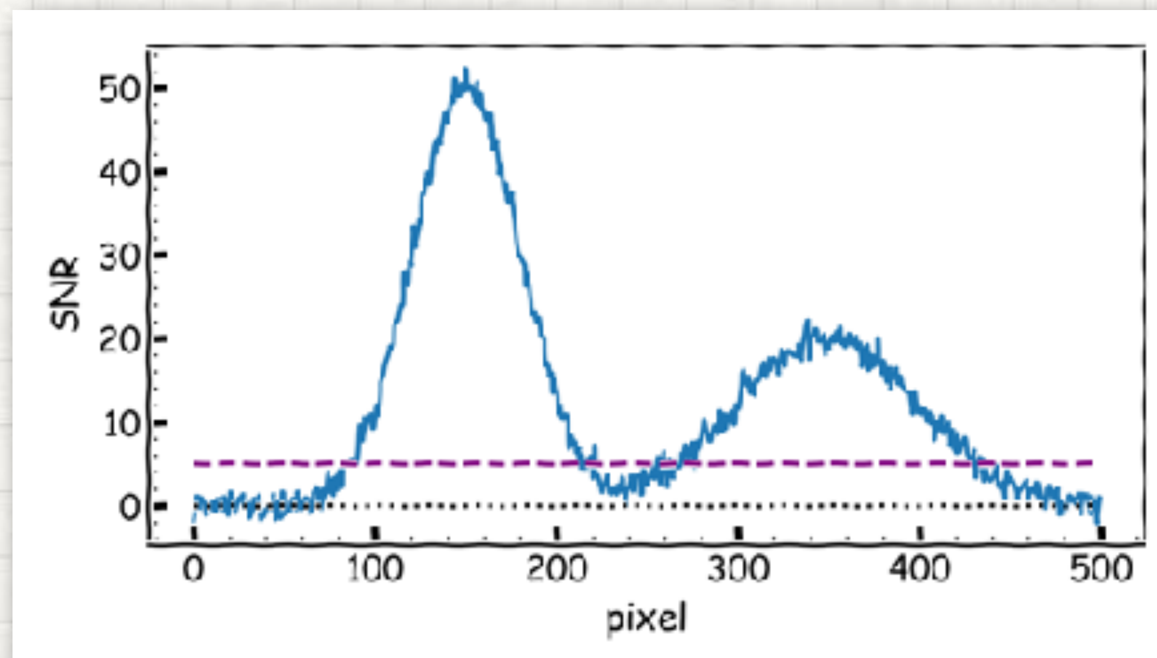

```
#overplot noise level contours on image
#contours are in steps of 4σ, starting at 1σ.


plt.figure(figsize = (16,8))
plt.subplot(131)
plt.imshow(hst_blue.data,origin = 'lower', vmax = np.percentile(hst_blue.data,99))

#zoom-in
cutout = Cutout2D(hst_blue.data, (332,270), (40,40))
cutout.plot_on_original(color='white')
ax2 = plt.subplot(132)
plt.imshow(cutout.data,origin = 'lower', vmax = np.percentile(hst_blue.data,99))
plt.contour(cutout.data,levels = hst_blue_noise_level*np.arange(1,100,4), colors = 'white')
plt.show()
```

STEP #1

HUNTING GALAXIES IN AN ASTRONOMICAL IMAGE

- generate signal-to-noise ratio (SNR) map
- define a threshold level and generate a mask



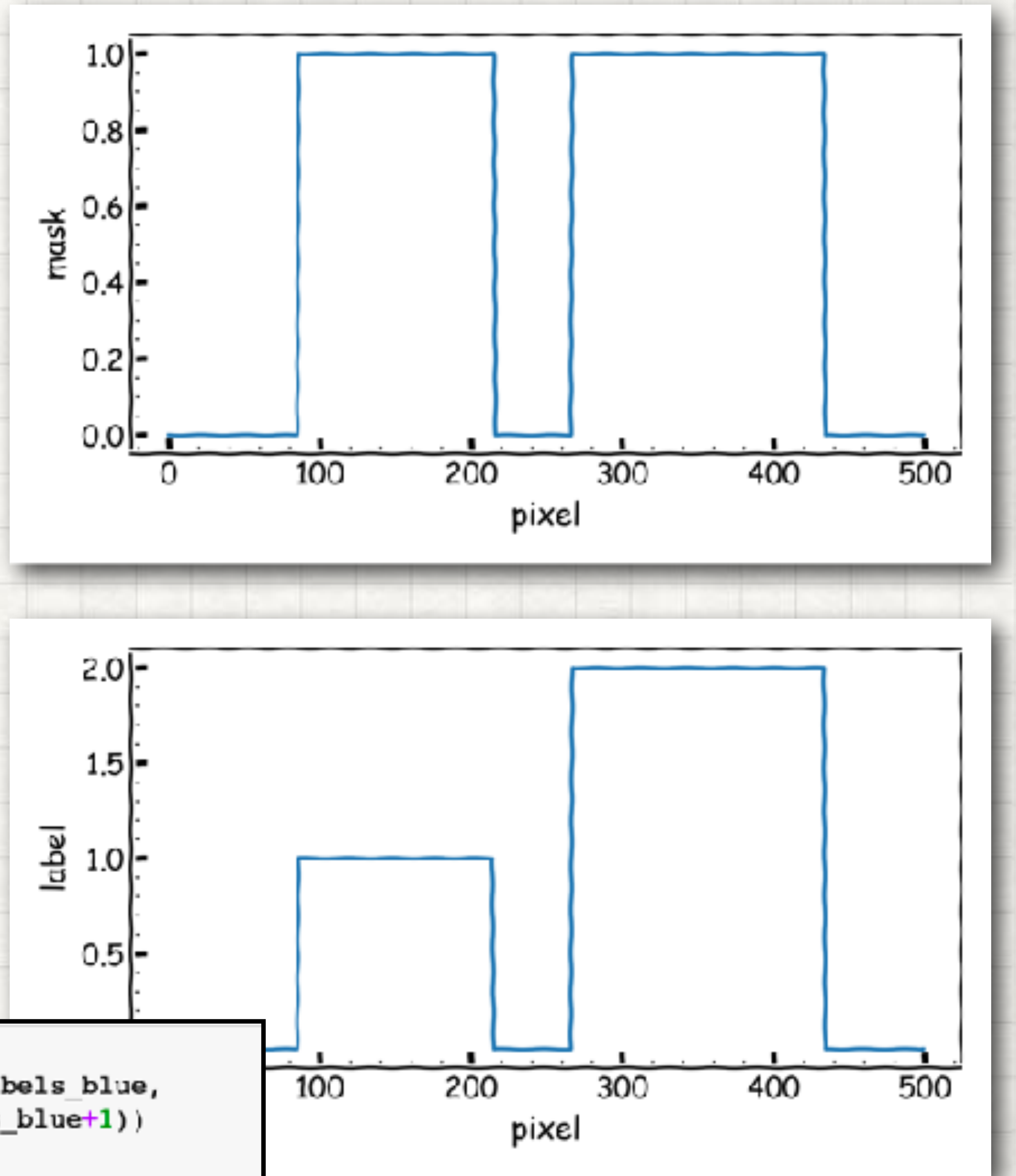



```
#find pixels above the threshold
mask_blue = hst_blue.data>(sn_threshold_blue*hst_blue_noise_level)
```

STEP #1

HUNTING GALAXIES IN AN ASTRONOMICAL IMAGE

- generate signal-to-noise ratio (SNR) map
- define a threshold level and generate a mask
- label the structures in a multidimensional array

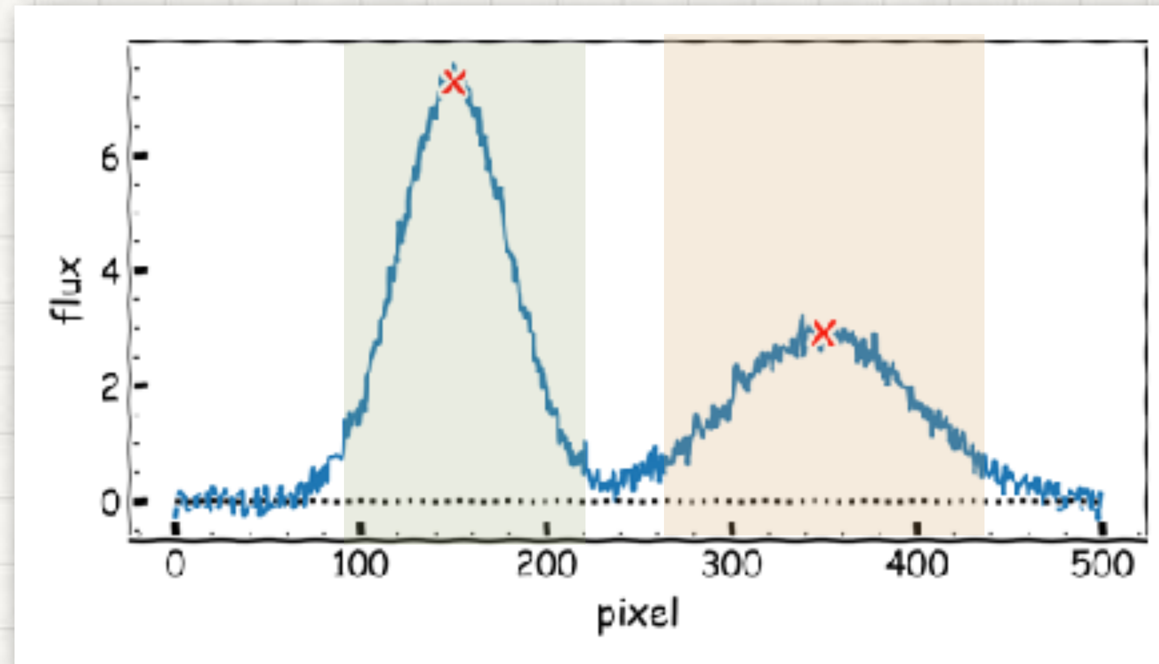


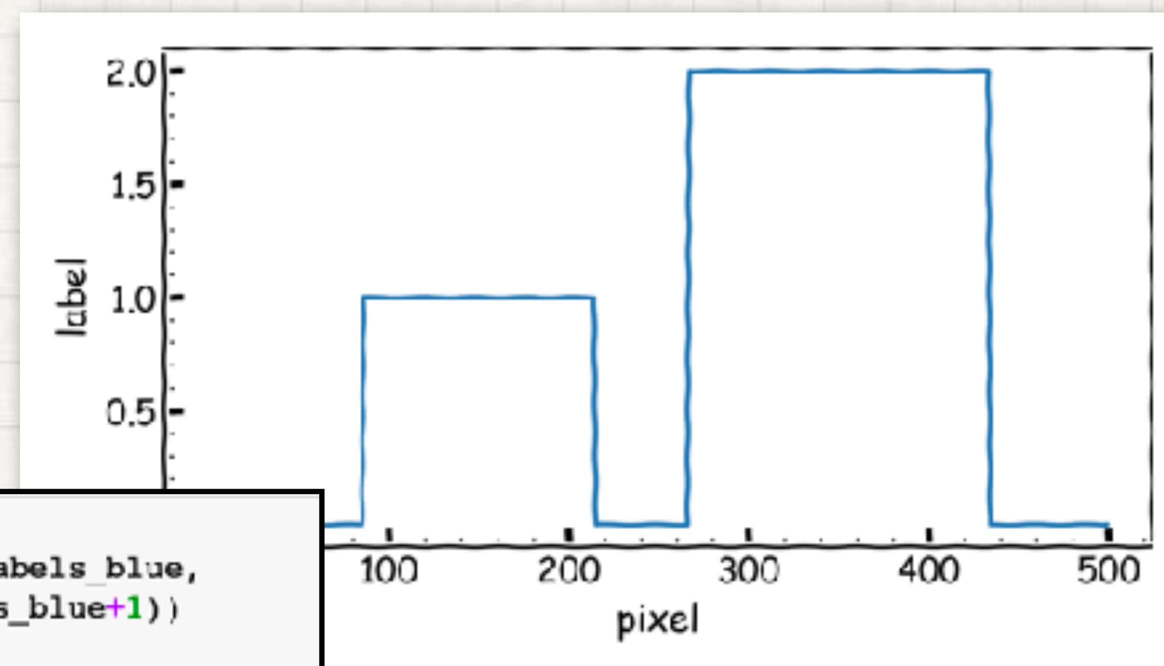

```
#determine the centroid of all identified objects
blue_pixels = ndimage.measurements.center_of_mass(hst_blue.data, map_labels_blue,
                                                  np.arange(1,n_labels_blue+1))
blue_pixels_arr = np.asarray(blue_pixels)

#turn pixel coordinates in to astronomical coordinates (x,y) -> (RA,Dec)
blue_coord = hst_blue_wcs.all_pix2world(blue_pixels_arr[:,1],blue_pixels_arr[:,0],0)
```

STEP #1

HUNTING GALAXIES IN AN ASTRONOMICAL IMAGE

- generate signal-to-noise ratio (SNR) map
- define a threshold level and generate a mask
- label the structures in a multidimensional array
- calculate the photometric centroid






```
#determine the centroid of all identified objects
blue_pixels = ndimage.measurements.center_of_mass(hst_blue.data, map_labels_blue,
                                                   np.arange(1,n_labels_blue+1))
blue_pixels_arr = np.asarray(blue_pixels)

#turn pixel coordinates in to astronomical coordinates (x,y) -> (RA,Dec)
blue_coord = hst_blue_wcs.all_pix2world(blue_pixels_arr[:,1],blue_pixels_arr[:,0],0)
```

STEP #1

HUNTING GALAXIES IN AN ASTRONOMICAL IMAGE

- generate signal-to-noise ratio (SNR) map
- define a threshold level and generate a mask
- label the structures in a multidimensional array
- calculate the photometric centroid
- generate a catalogue of extragalactic sources

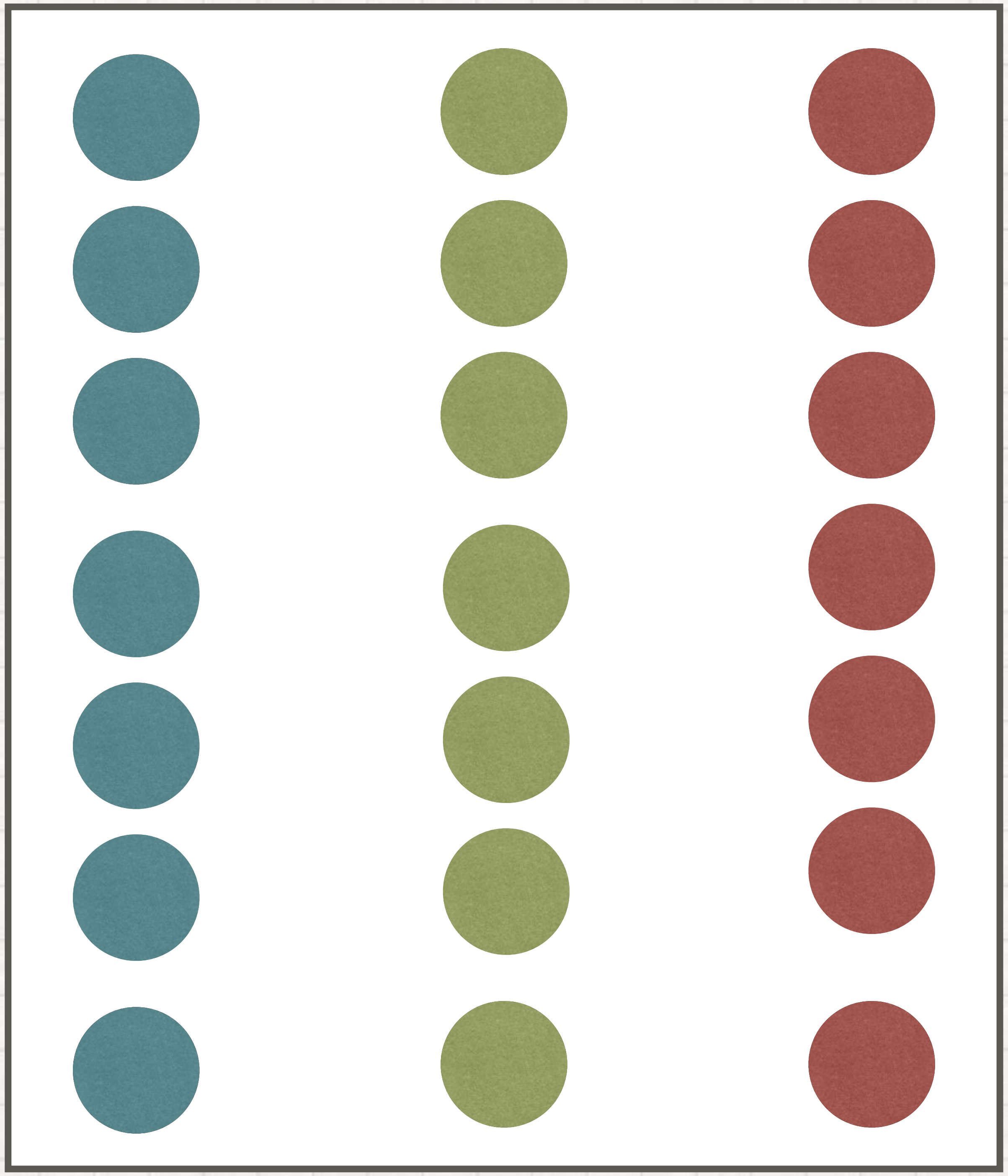
```python
#determine the centroid of all identified objects
blue_pixels = ndimage.measurements.center_of_mass(hst_blue.data, map_l
                                                  np.arange(1,n_label
blue_pixels_arr = np.asarray(blue_pixels)

#turn pixel coordinates in to astronomical coordinates (x,y) -> (RA,Dec)
blue_coord = hst_blue_wcs.all_pix2world(blue_pixels_arr[:,1],blue_pixels_arr[:,0],0)
```

```
#RA                      DEC
1.501201533323518902e+02 2.255580432211921504e+00
1.501271569594991888e+02 2.255662676511354547e+00
1.501260879294809740e+02 2.255729704911888067e+00
1.501245893425685267e+02 2.255992453901231709e+00
1.501359065662555281e+02 2.256141202033094206e+00
1.501328519825382273e+02 2.256086014050021404e+00
1.501188178368863362e+02 2.256400657842178692e+00
1.501307379875519246e+02 2.256660074344509326e+00
1.501328057471069712e+02 2.256786633523686803e+00
1.501175604341250676e+02 2.256863299909120268e+00
1.501175925696436195e+02 2.257121885978219833e+00
1.501234895518483938e+02 2.257183293693242110e+00
1.501179540543875248e+02 2.257425053648479896e+00
1.501181811289956158e+02 2.257252776207845812e+00
1.501303859047442586e+02 2.257374631758512162e+00
1.501326013344514649e+02 2.257497129142929548e+00
1.501177491970632332e+02 2.257863616106043914e+00
1.501358683832628174e+02 2.257885989371087643e+00
1.501246923072806396e+02 2.257919808742051782e+00
1.501348574272848566e+02 2.258186387017588004e+00
1.501299356961779097e+02 2.258206235468105838e+00
1.501284924459576757e+02 2.258283152452662090e+00
1.501231468347557723e+02 2.258997788900573234e+00
1.501227380881309728e+02 2.258947759150648160e+00
1.501283346338818205e+02 2.258976759346065233e+00
1.501273777755998537e+02 2.259266086072532875e+00
1.501212778608782230e+02 2.259292784996972614e+00
1.501303071086194905e+02 2.259624996826600452e+00
1.501346842865602014e+02 2.259544197481857974e+00
1.501358460453445218e+02 2.259968576923834682e+00
1.501316709855317697e+02 2.260611770436528278e+00
1.501235582304366574e+02 2.260685561946683464e+00
1.501199746894155282e+02 2.261033654606239729e+00
1.501277998341232944e+02 2.261113545193483176e+00
1.501230932014763937e+02 2.261562707080464474e+00
```

STEP #2

SEARCHING GALAXIES AT Z>6 BY USING CATALOGUES

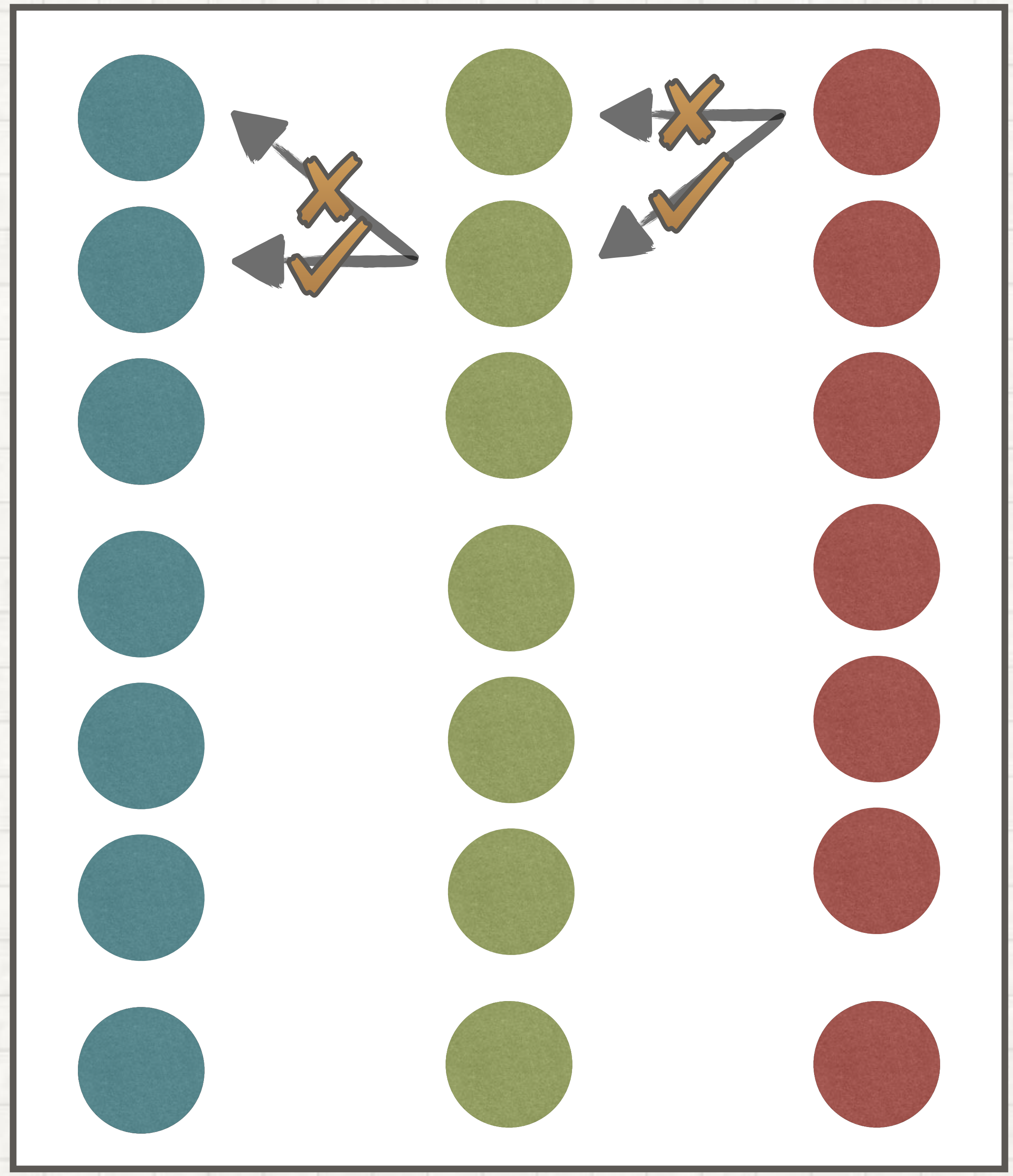
- cross-correlation of galaxy catalogues
- galaxies at z>6 are in the `red' and `green' catalogue, but they are not included in the `blue' catalogue



STEP #2

SEARCHING GALAXIES AT Z>6 BY USING CATALOGUES

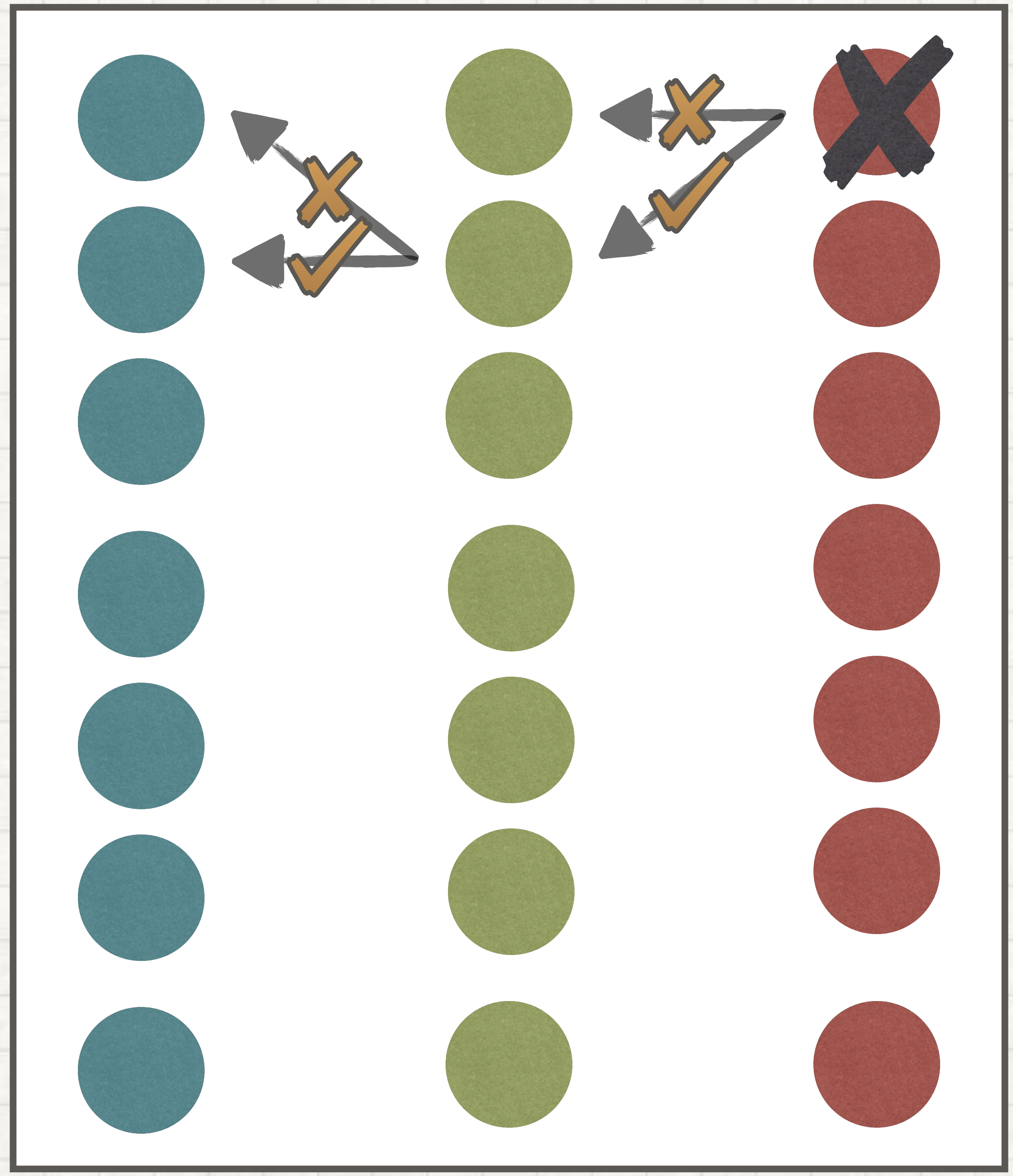
- cross-correlation of galaxy catalogues
- galaxies at z>6 are in the `red' and `green' catalogue, but they are not included in the `blue' catalogue



STEP #2

SEARCHING GALAXIES AT Z>6 BY USING CATALOGUES

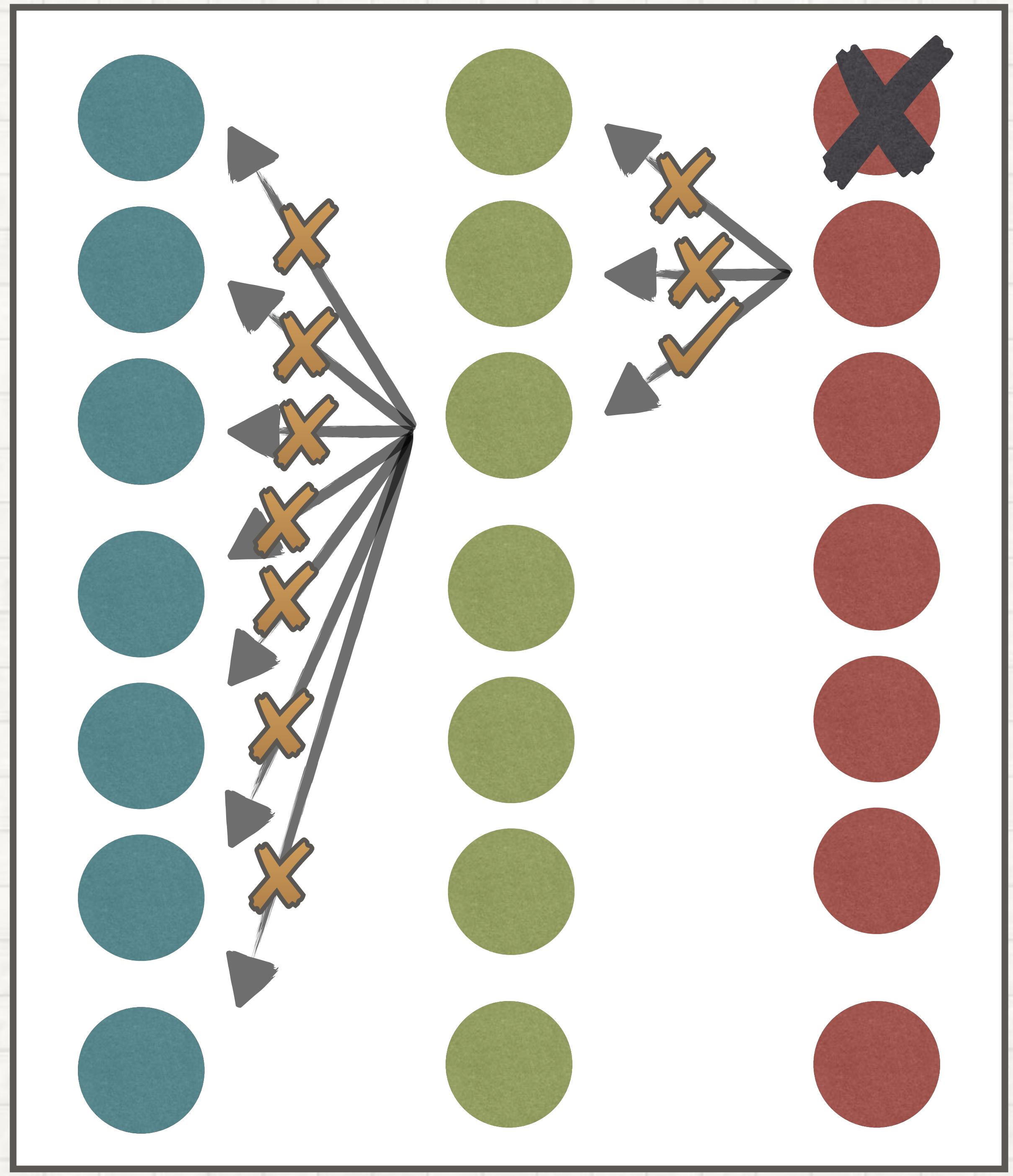
- cross-correlation of galaxy catalogues
- galaxies at z>6 are in the `red' and `green' catalogue, but they are not included in the `blue' catalogue



STEP #2

SEARCHING GALAXIES AT Z>6 BY USING CATALOGUES

- cross-correlation of galaxy catalogues
- galaxies at z>6 are in the `red' and `green' catalogue, but they are not included in the `blue' catalogue

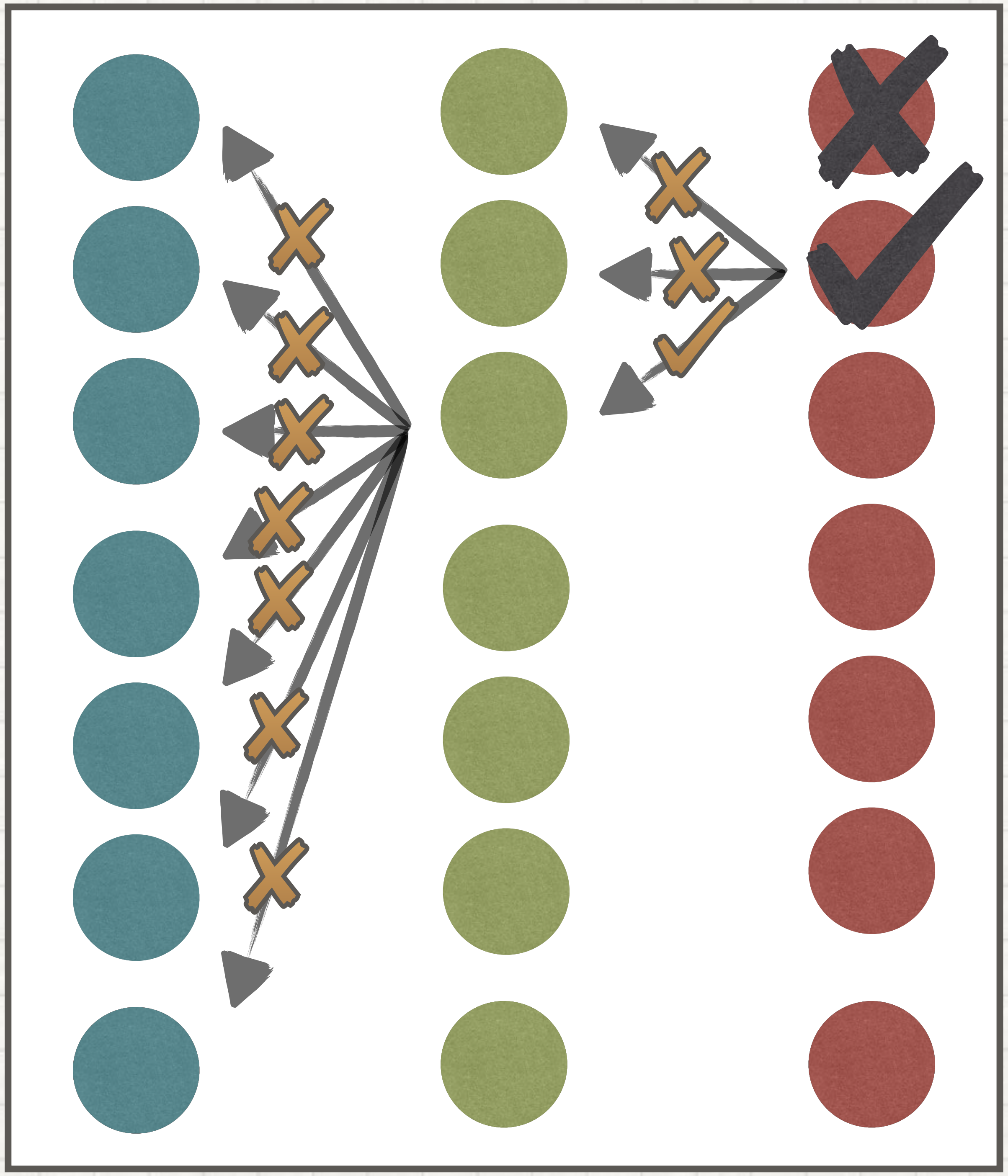


STEP #2

SEARCHING GALAXIES AT Z>6 BY USING CATALOGUES

- cross-correlation of galaxy catalogues

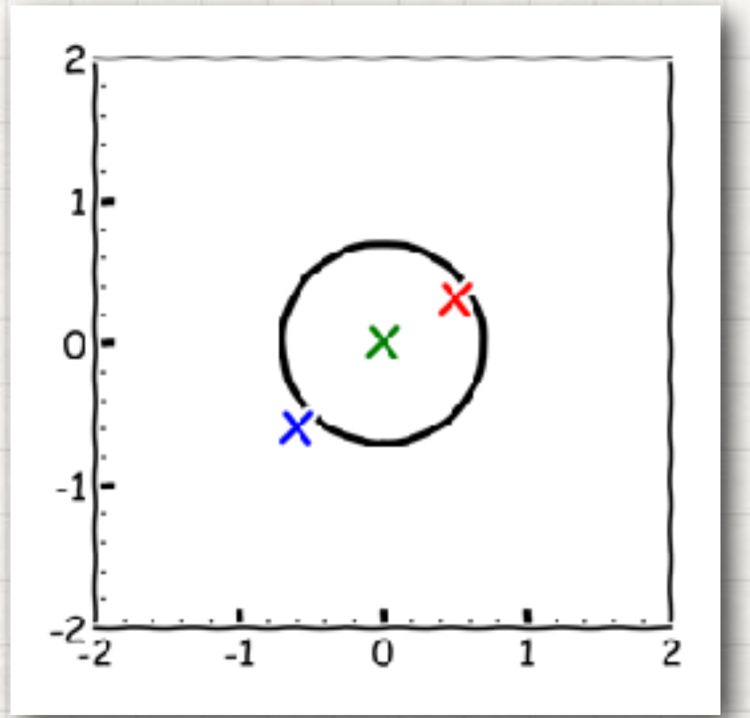
- galaxies at z>6 are in the `red' and `green' catalogue, but they are not included in the `blue' catalogue

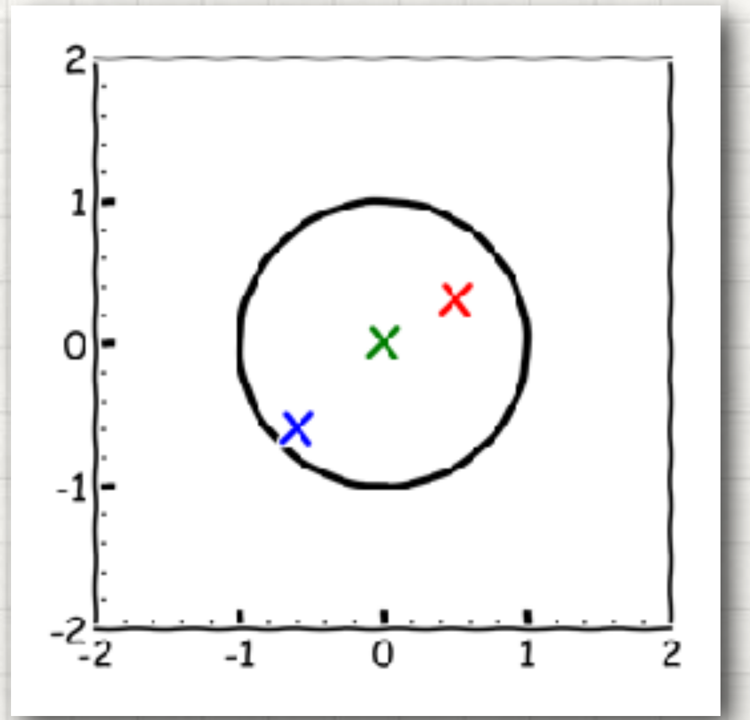


STEP #2

SEARCHING GALAXIES AT Z>6 BY USING CATALOGUES

- cross-correlation of galaxy catalogues
- galaxies at z>6 are in the `red' and `green' catalogue, but they are not included in the `blue' catalogue
- determine the astrometry uncertainty
- define a area around my candidate galaxy at z>6

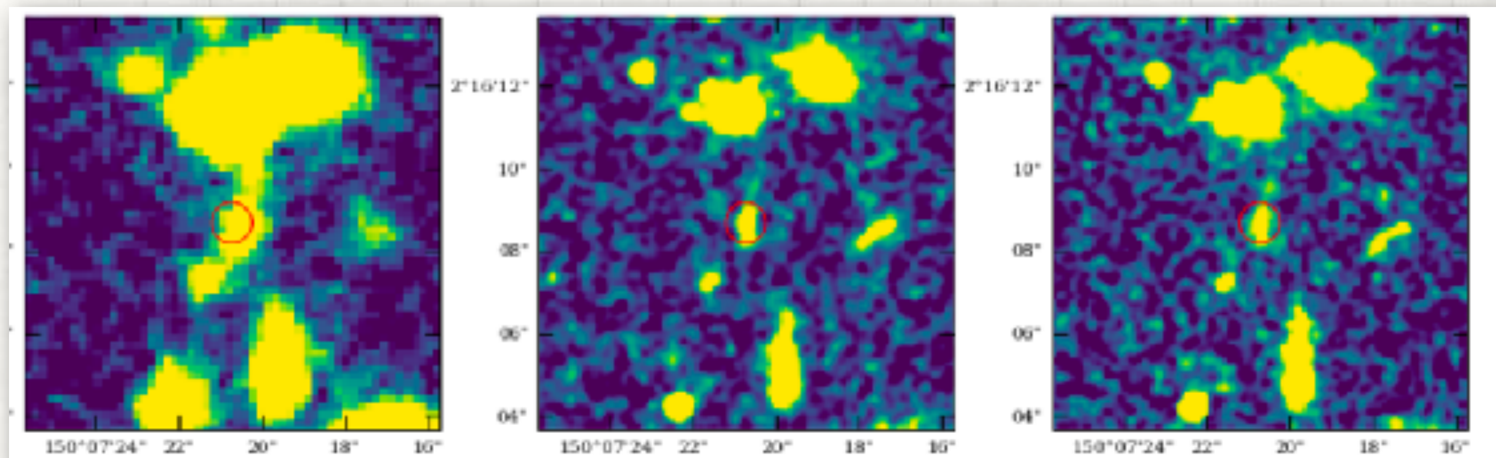


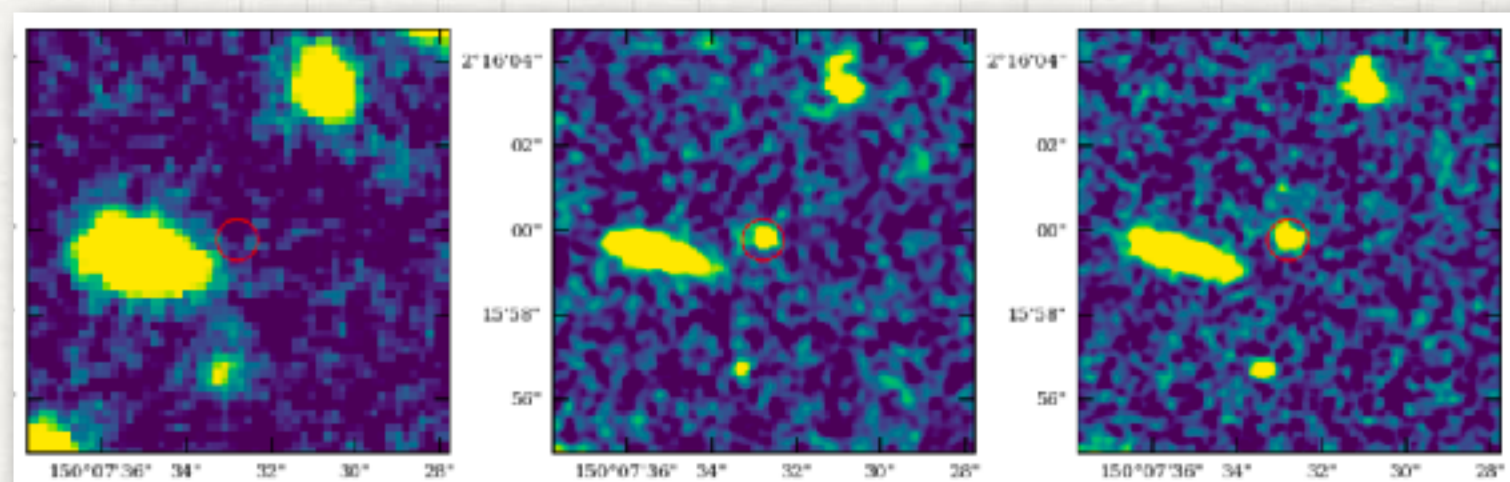


STEP #2

SEARCHING GALAXIES AT Z>6 BY USING CATALOGUES

- cross-correlation of galaxy catalogues
- galaxies at z>6 are in the `red' and `green' catalogue, but they are not included in the `blue' catalogue
- determine the astrometry uncertainty
- define a area around my candidate galaxy at z>6
- visual inspection

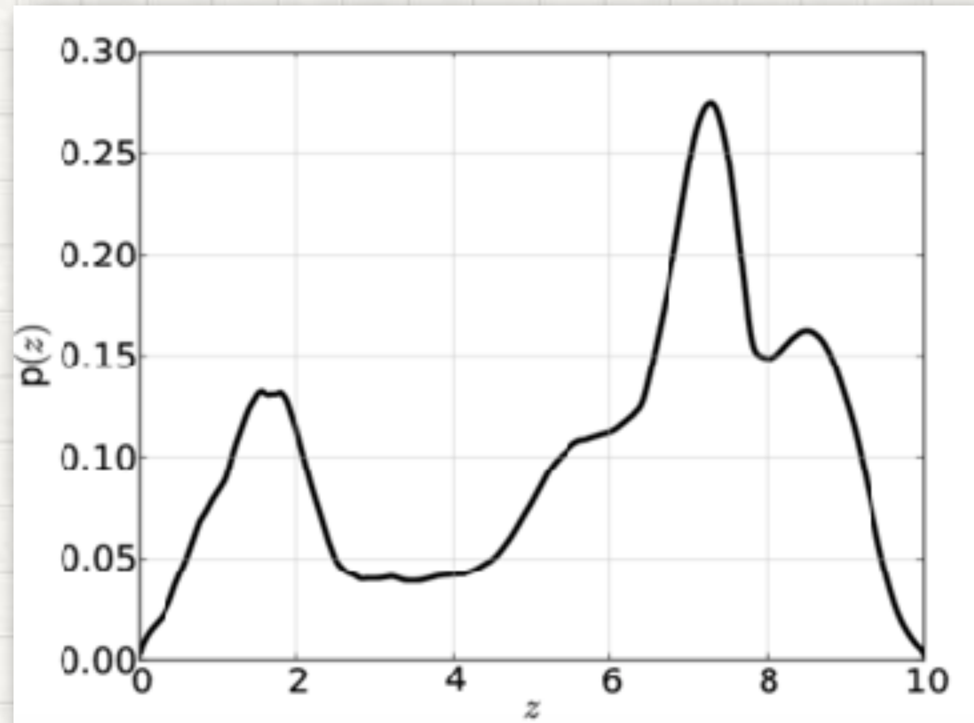


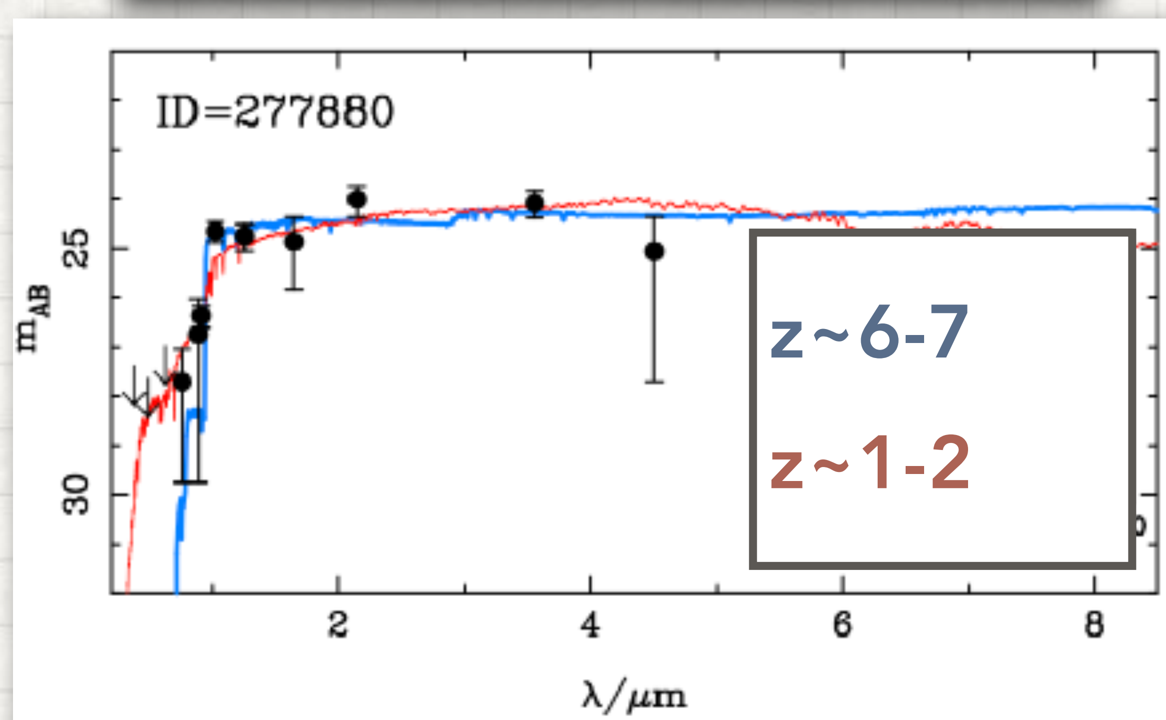


STEP #3

CONFIRMING THE Z OF OUR CANDIDATES AND …

- Lyman Breck Technique: large uncertainty on *z* determination
- The detection of two emission line is necessary to confirm the redshift of our candidate





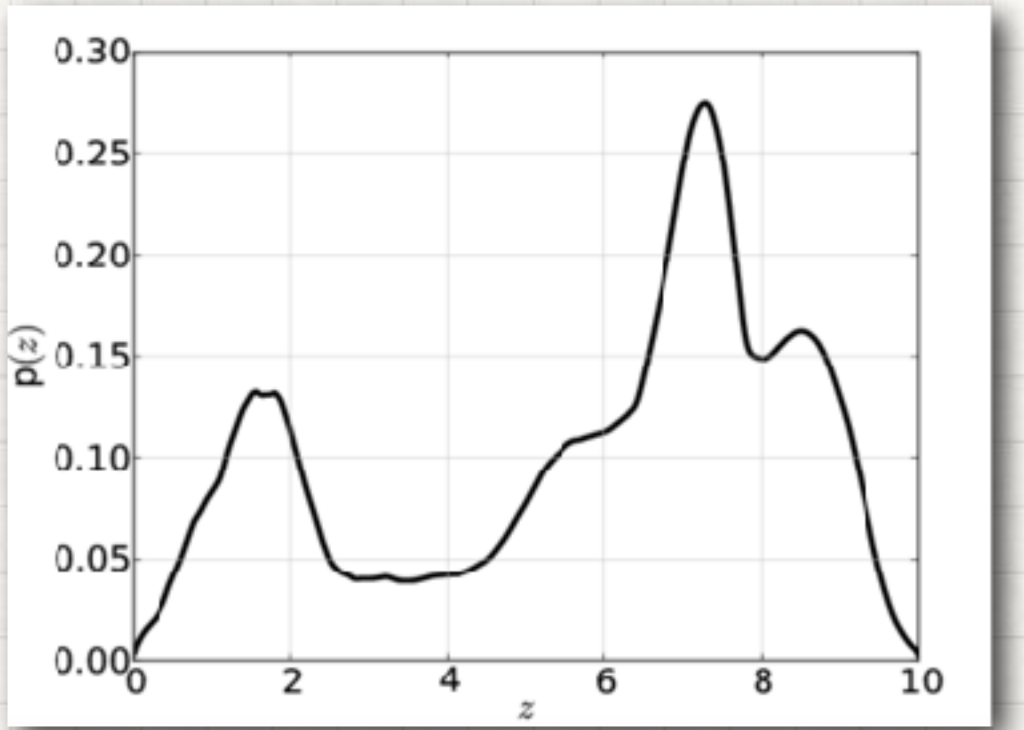
STEP #3

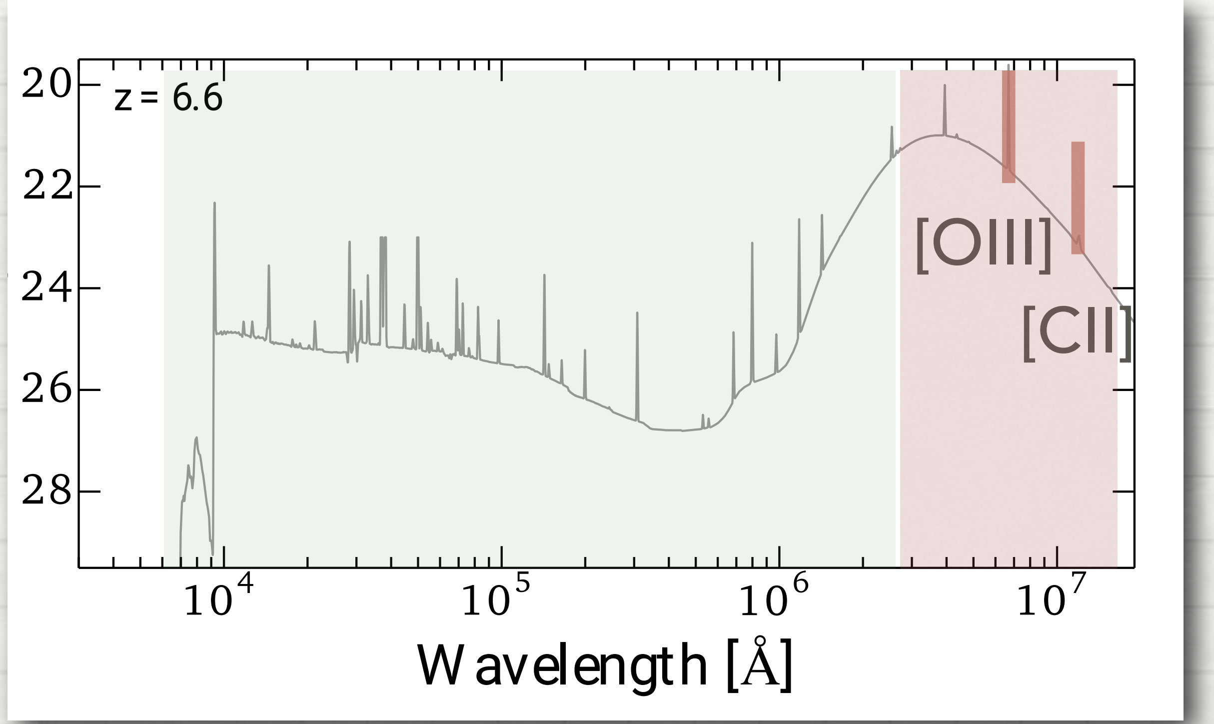
CONFIRMING THE Z OF OUR CANDIDATES AND …

- Lyman Breck Technique: large uncertainty on *z* determination

- The detection of two emission line is necessary to confirm the redshift of our candidate

- Our ALMA observations target [CII]158µm and [OIII]88µm, which are redshifted to mm (~300-500 GHz) bands at z≳6

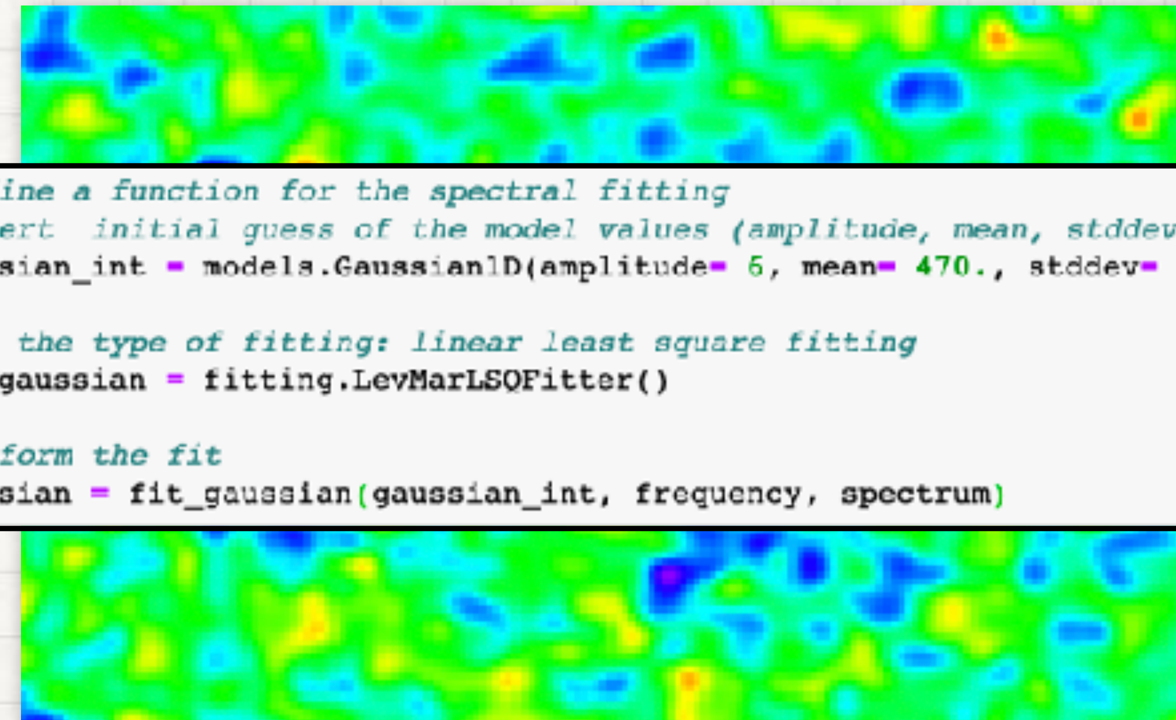




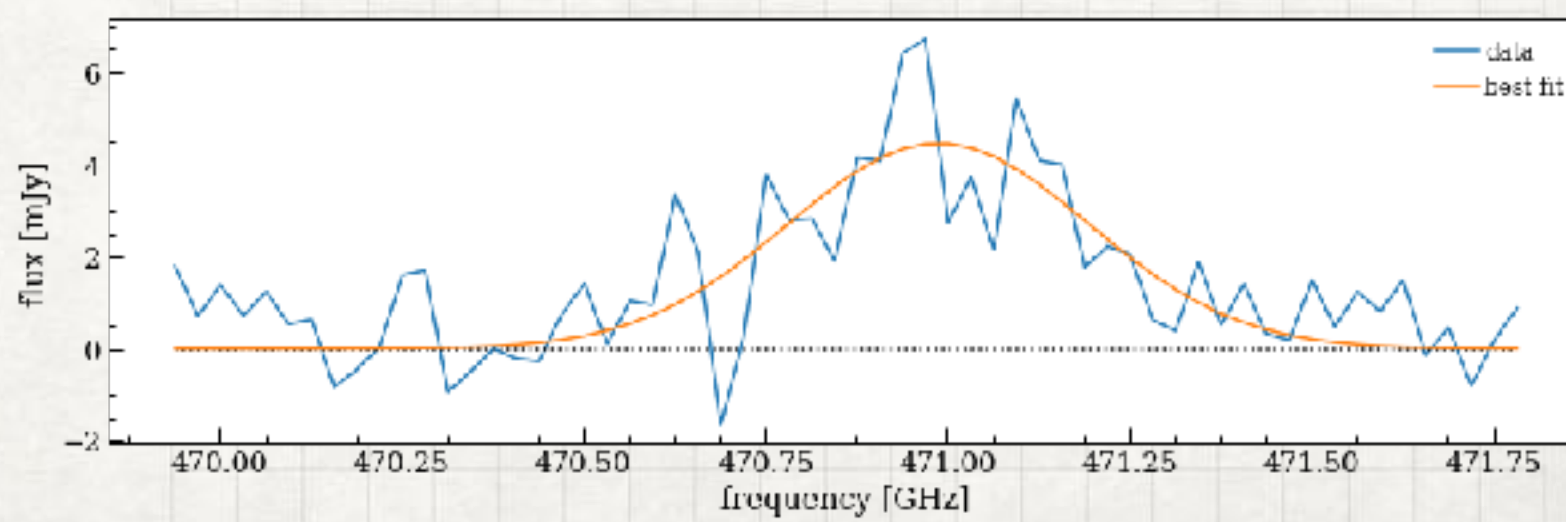
STEP #3

CONFIRMING THE Z OF OUR CANDIDATES AND …

- Lyman Breck Technique: large uncertainty on *z* determination
- The detection of two emission line is necessary to confirm the redshift of our candidate
- Our ALMA observations target [CII]158μm and [OIII]88μm, which are redshifted to mm (~300-500 GHz) bands at z≳6
- Extract a spectrum from our cube
- Fit the spectrum with a Gaussian profile



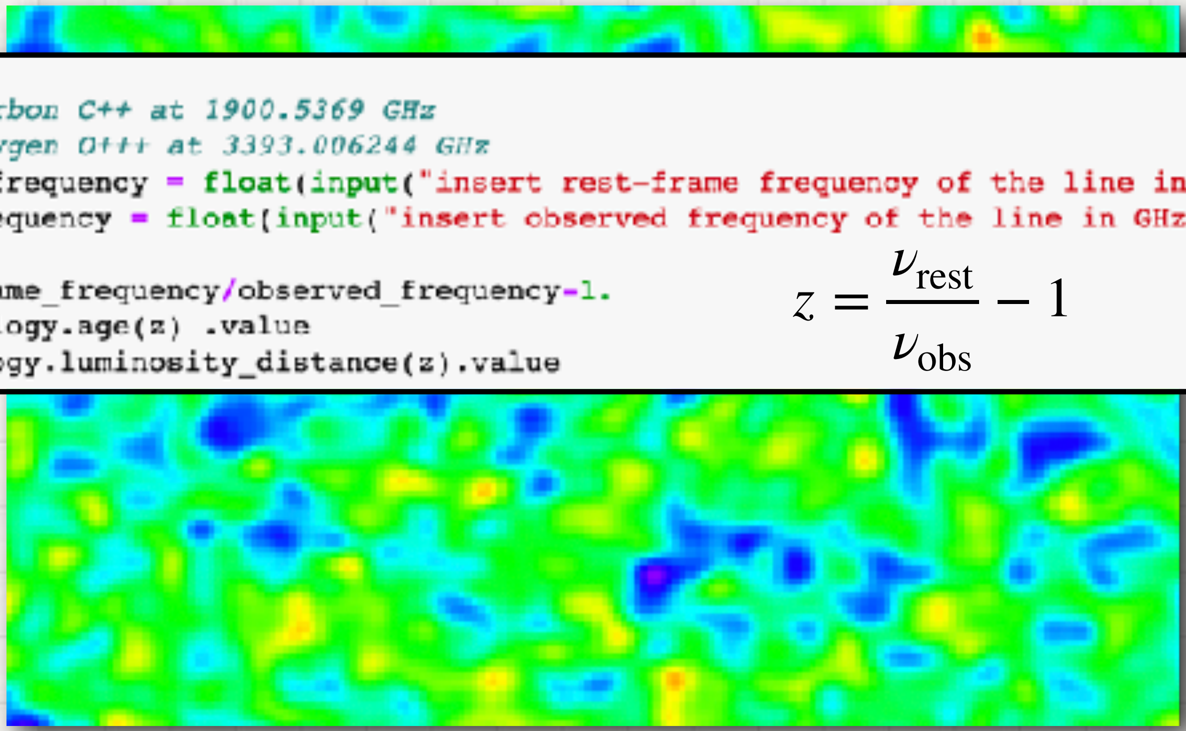
```
#define a function for the spectral fitting
#insert  initial guess of the model values (amplitude, mean, stddev)
gaussian_int = models.Gaussian1D(amplitude= 5, mean= 470., stddev= 0.5)

#set the type of fitting: linear least square fitting
fit_gaussian = fitting.LevMarLSQFitter()

#perform the fit
gaussian = fit_gaussian(gaussian_int, frequency, spectrum)
```



STEP #3

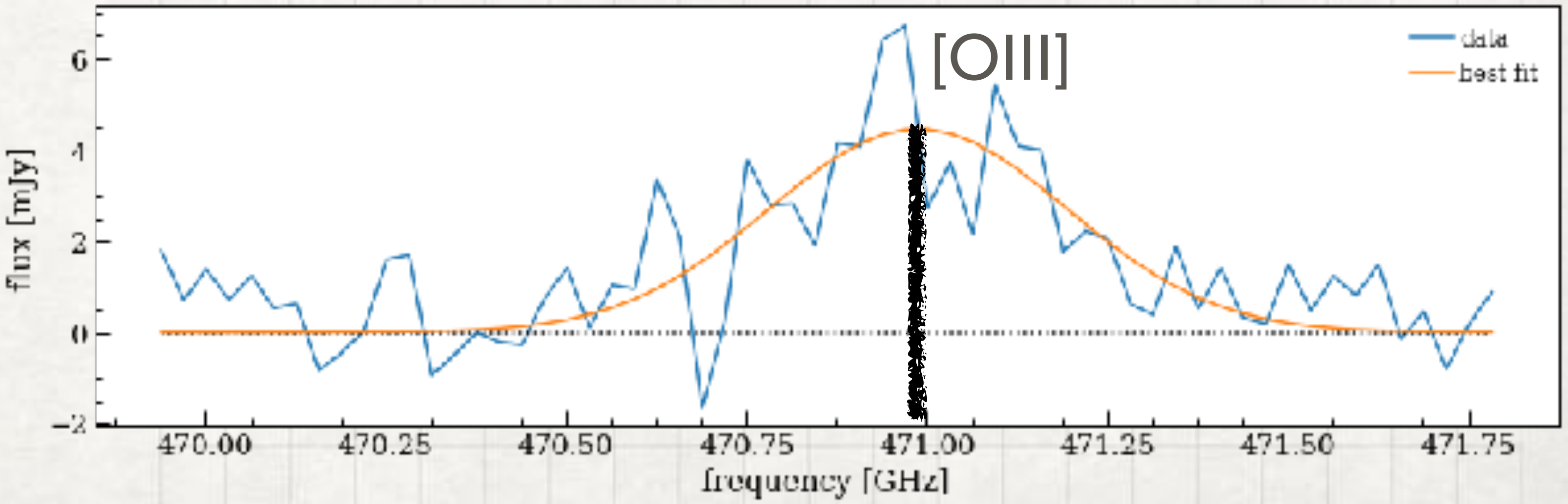
CONFIRMING THE Z OF OUR CANDIDATES AND …

- Lyman Breck Technique: large uncertainty on *z* determination
- The detection of two emission line is necessary to confirm the redshift of our candidate
- Our ALMA observations target [CII]158μm and [OIII]88μm, which are redshifted to mm (~300-500 GHz) bands at z≳6
- Extract a spectrum from our cube
- Fit the spectrum with a Gaussian profile
- Estimate redshift



```
#suggestion
#Ionized carbon C++ at 1900.5369 GHz
#Ionized oxygen O+++ at 3393.006244 GHz
rest_frame_frequency = float(input("insert rest-frame frequency of the line in GHz: "))
observed_frequency = float(input("insert observed frequency of the line in GHz: "))

z = rest_frame_frequency/observed_frequency-1.
age = cosmology.age(z) .value
dl = cosmology.luminosity_distance(z).value
```

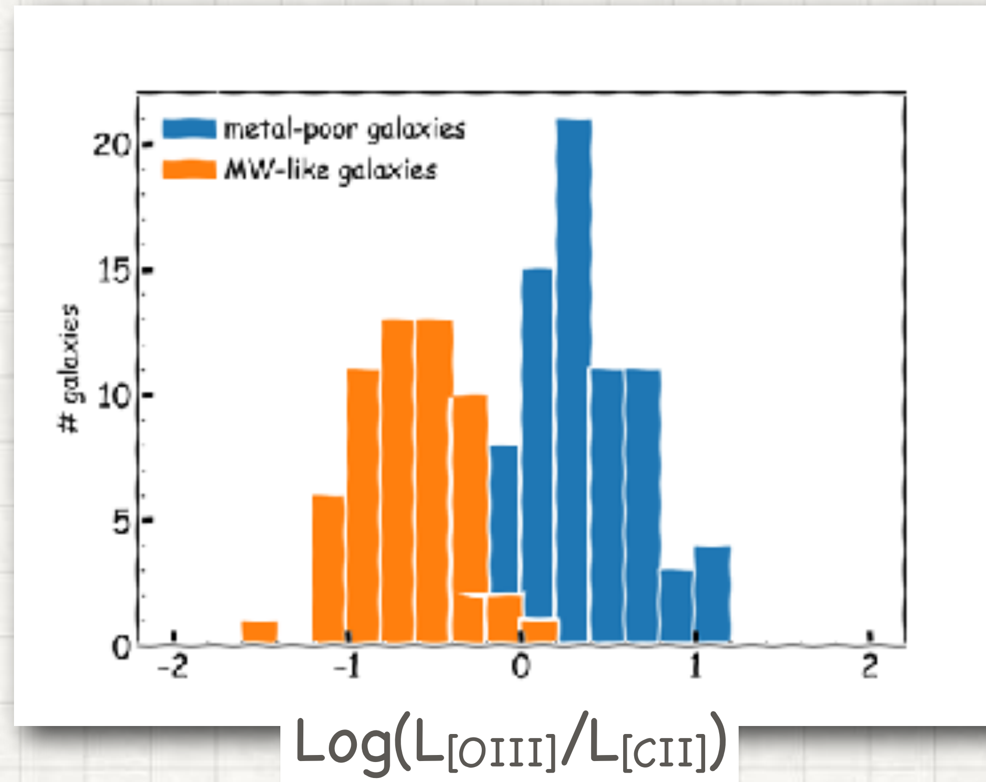
$$z = \frac{\nu_{\rm rest}}{\nu_{\rm obs}} - 1$$



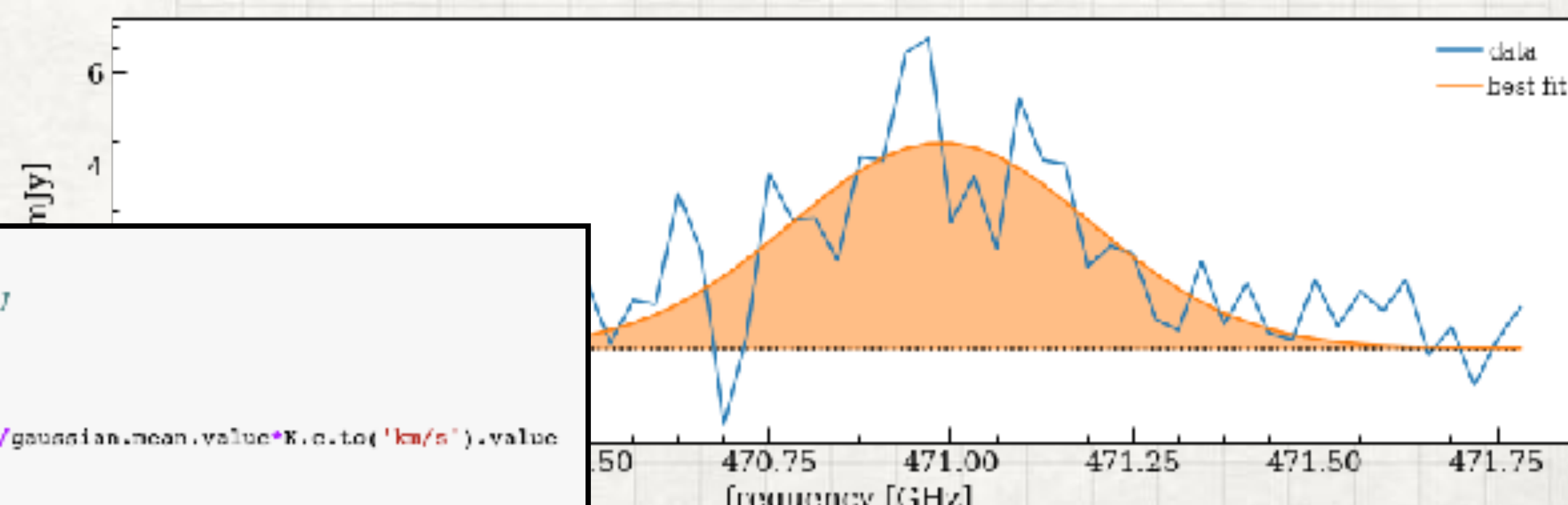
STEP #3

… DETERMINING GALAXY PROPERTIES

- [CII]158µm and [OIII]88µm line can be used to distinguish a Milky Way like galaxy from a metal-poor galaxy (i.e. low heavy element abundance)
- Estimate the luminosity of the [CII]158µm and [OIII]88µm line
- Determine the luminosity ratio between the two lines



$$L_{\rm line} = 4\pi D_{\rm L}^{2} \int S_{\nu} d\nu$$

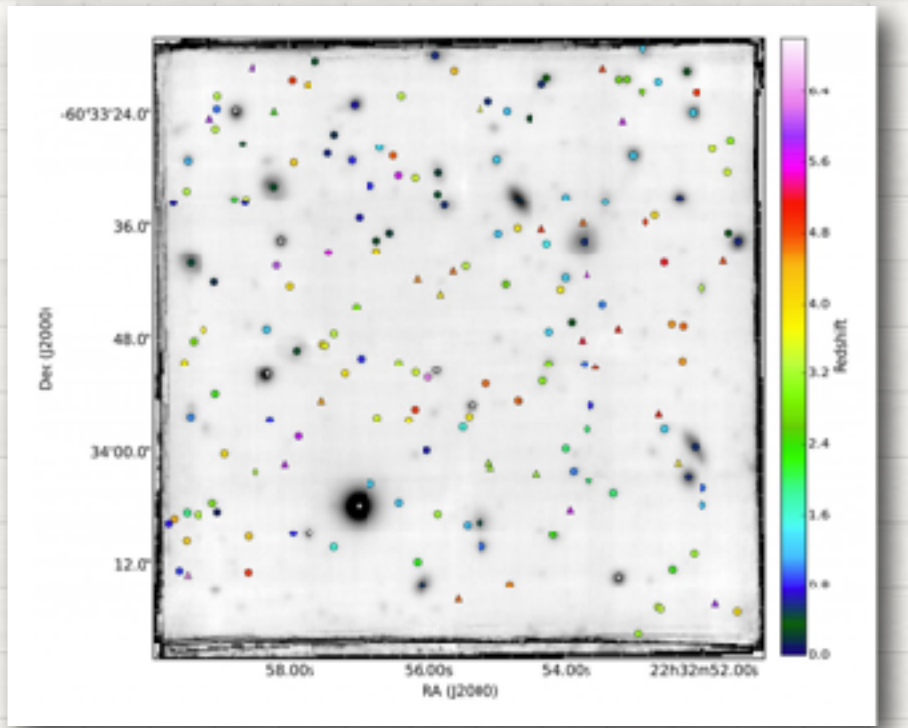


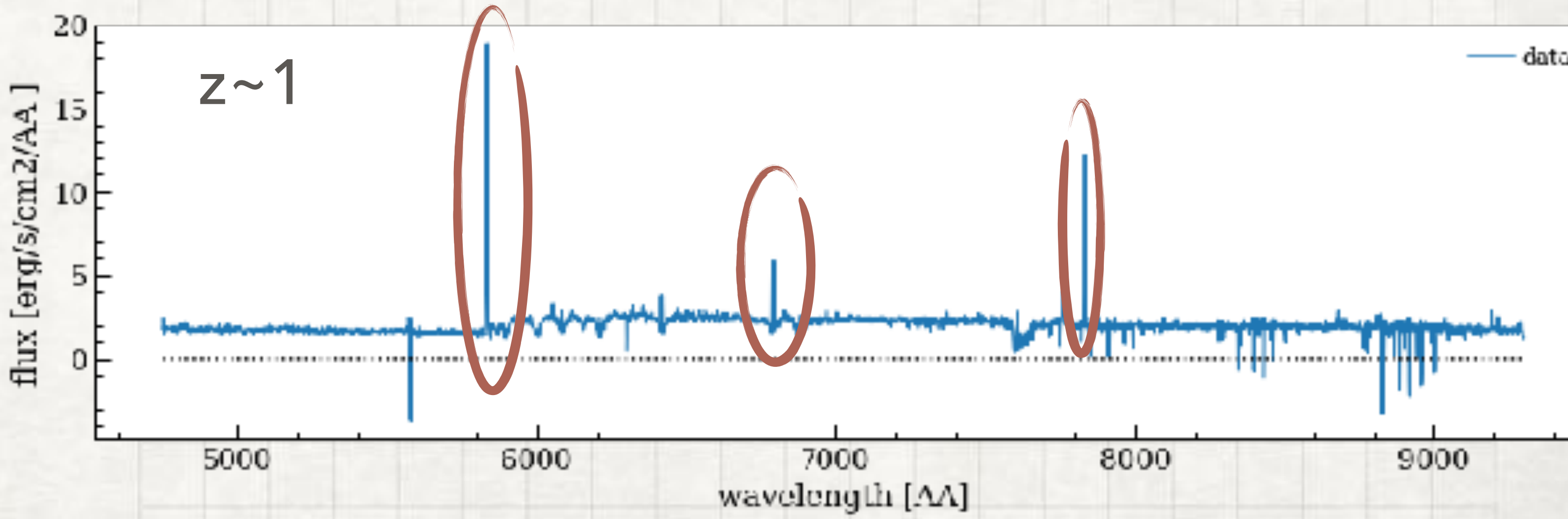

```
#estimate the luminosity of the line
#equation for far-infrared lines
# L = 1.04 x 10**-3 x Sdv[Jy km/s] x dl**2[Mpc**2] x observed_frequency[GHz]
# where
# Sdv is the integrated flux of the line
# Sdv = amplitude x (2 x pi)**0.5 x stddev / mean x c[velocity speed]

Sdv = 1e-3*gaussian.amplitude.value*gaussian.stddev.value*np.sqrt(2.*np.pi)/gaussian.mean.value*K.c.to('km/s').value
print("Sdv = {} Jy km/s".format(Sdv))


L = 1.04e-3*Sdv*dl**2*observed_frequency/1e8
print("L = {} 10^8 Lsun".format(L))
```

STEP #4

A MORE COMPLEX 3D DATACUBE

- MUSE (Multi Unit Spectroscopic Explorer) observations at 4700-9100Å
- >200 galaxies out to z~6.5
- low-z galaxies (z~1-3) have multiple rest-frame optical emission lines





STEP #4

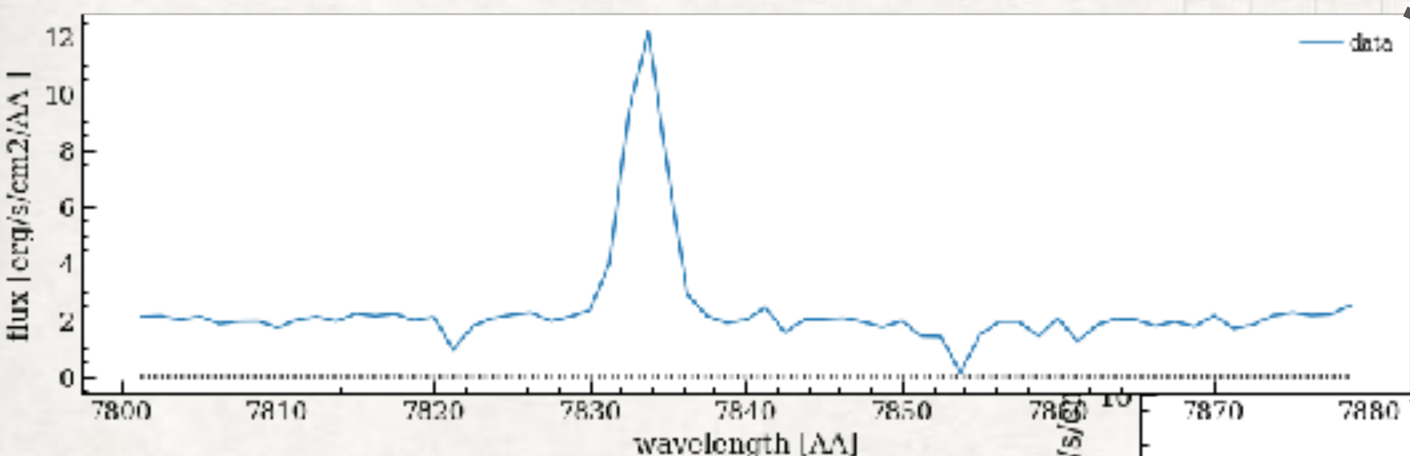
A MORE COMPLEX 3D DATACUBE

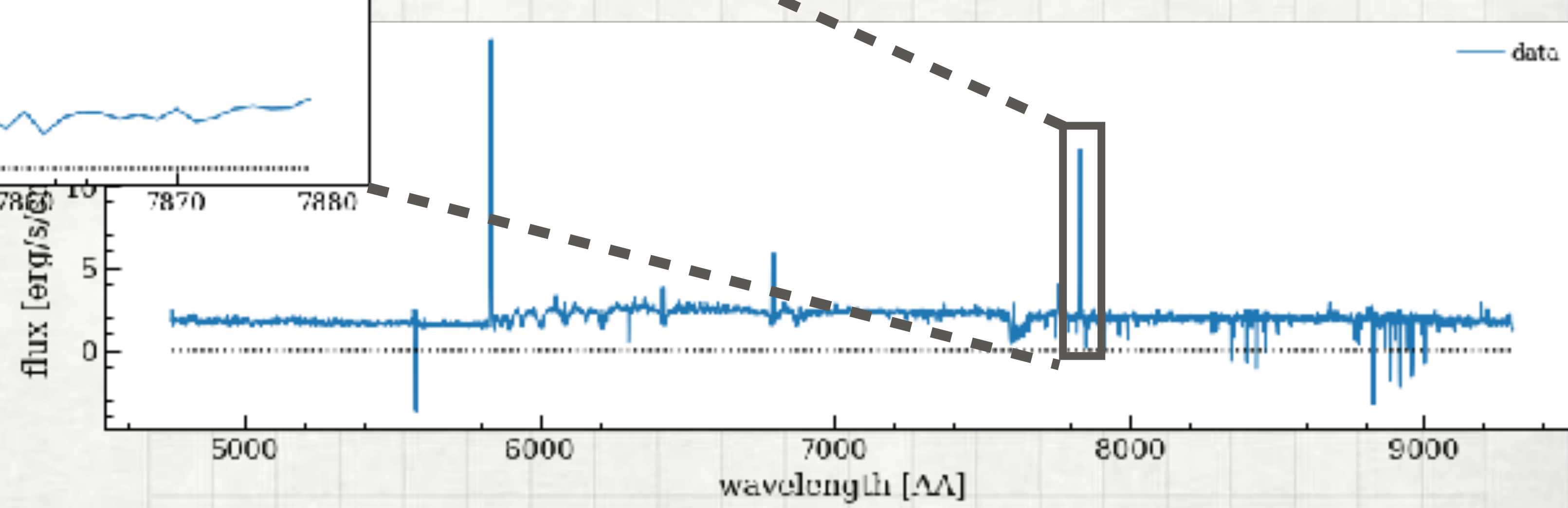
- MUSE (Multi Unit Spectroscopic Explorer) observations at 4700-9100Å

- >200 galaxies out to z~6.5

- low-z galaxies (z~1-3) have multiple rest-frame optical emission lines

```
#select a region to zoom-in
wl_1 = float(input("insert a wavelength min: "))
wl_2 = float(input("insert a wavelength max: "))


wl_crop = wavelength[np.where((wavelength>wl_1) & (wavelength<wl_2))]
spec_crop = spectrum[np.where((wavelength>wl_1) & (wavelength<wl_2))]

plt.figure(figsize = (12,4))
plt.plot(wl_crop, spec_crop, label = 'data')
plt.plot(wl_crop,wl_crop*0,':',color = 'black')
plt.xlabel('wavelength [AA]')
plt.ylabel('flux [erg/s/cm2/AA ]')
plt.legend()
plt.show()
```



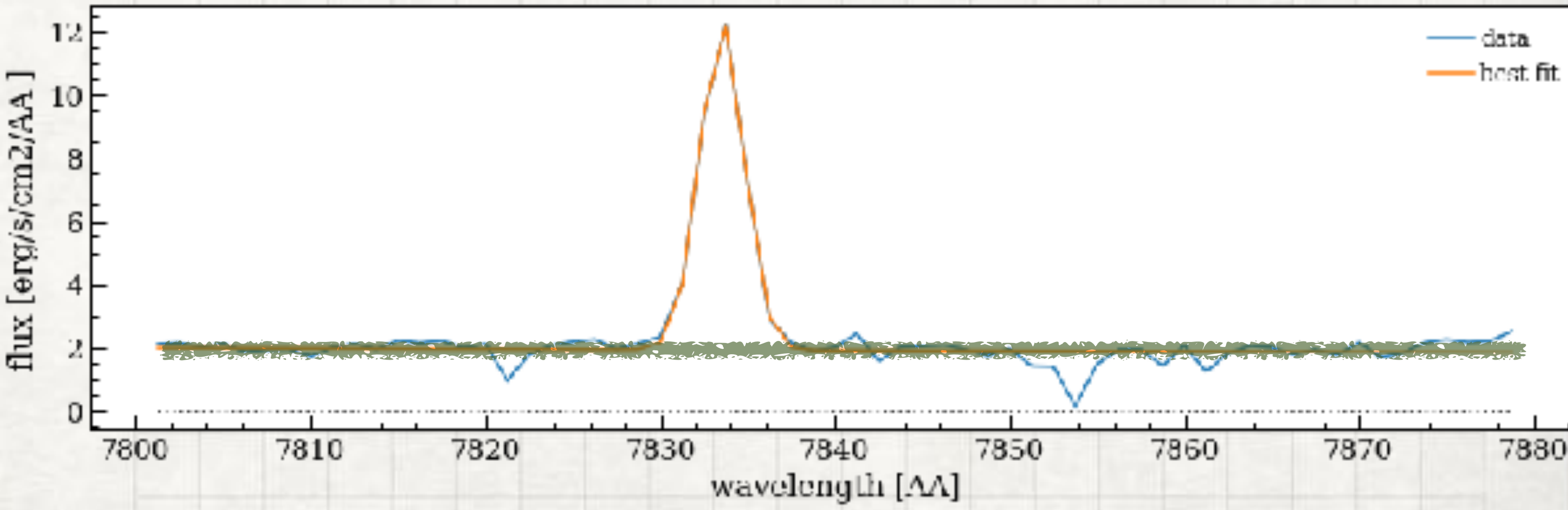


STEP #4

A MORE COMPLEX 3D DATACUBE

- MUSE (Multi Unit Spectroscopic Explorer) observations at 4700-9100Å
- >200 galaxies out to z~6.5
- low-z galaxies (z~1-3) have multiple rest-frame optical emission lines
- a continuum emission due to the stellar population

```
#define a function for the spectral fitting
#insert  initial guess of the model values (amplitude, mean, stddev)
gaussian_int = models.Gaussian1D(amplitude=15, mean=7835, stddev=1)
polynomial_init = models.Linear1D()

#set the type of fitting: linear least square fitting
fit_gaussian = fitting.LevMarLSQFitter()

#perfome the fit
gaussian = fit_gaussian(gaussian_int+polynomial_init, wl_crop, spec_crop)
```



STEP #4

A MORE COMPLEX 3D DATACUBE

- MUSE (Multi Unit Spectroscopic Explorer) observations at 4700-9100Å

- >200 galaxies out to z~6.5

- low-z galaxies (z~1-3) have multiple rest-frame optical emission lines

- a continuum emission due to the stellar population

- series of possible of optical lines

```
#estimate the redshift (z) of the galaxy

#brightest lines in the UV and optical spectrum
#Lya 1215.67
#[OII]  3727.09 AA
#Hbeta  4862.69 AA
#Halpha 6564.61
#[OIII]a 5008.24
#[OIII]b 4960.30
#[NII]a 6549.84
#[NII]b 6585.23

list_restframe_wl = np.asarray([1215.67,3727.09,4862.69,6564.61,5008.24,4960.30,6549.84,6585.23])
list_restframe_wl_name = np.asarray(['Lya','[OII]','Hbeta','Halpha','[OIII]a','[OIII]b',
                                    '[NII]a','[NII]b'])

# rest_frame_wavelength = float(input("insert rest-frame wavelength of the line in AA: "))
observed_wavelength = float(input("insert observed wavelength of the line in AA: "))

for i in range(len(list_restframe_wl)):
    print(list_restframe_wl_name[i])
    z = observed_wavelength/list_restframe_wl[i]-1.
    print("redshift of the galaxy is: {}".format(z))
```